NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

A MEDIAL POINT TRANSFORMATION OF TWO-DIMENSIONAL DATA WITH APPLICATIONS TO DATA COMPRESSION AND NOISE REDUCTION

CRES Report No. 118-1

By

Irvin Edward Youngberg
Remote Sensing Laboratory
University of Kansas
Lawrence, Kansas

August 1967

Prepared by the University of Kansas Center for Research, Inc. Engineering Science Division, Lawrence, Kansas for the National Aeronautics and Space Administration (NASA) under NASA Contract No. NAS9-7175

MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

A MEDIAL POINT TRANSFORMATION OF TWO-DIMENSIONAL DATA WITH APPLICATIONS TO DATA COMPRESSION AND NOISE REDUCTION

by

Irvin Edward Youngberg

CRES Report #118-1

The Remote Sensing Laboratory
Information Sciences Group

Supported By:
NASA Contract NAS 9-7175

PRECEDING PAGE BLANK NOT FILMED.

ACKNOWLEDGEMENT

I am most grateful to my advisor, Dr. G. Kelly, for his assistance and helpful criticism.

I also wish to acknowledge the financial support of the National Aeronautics and Space Administration, under contract NSR 17-004-003, which made this work possible.

I further appreciate the assistance of Mrs. Nancy Farrell, Miss Ruth Ann Browne, and Mrs. Sylvia Dulin in preparing this paper.

I would also like to acknowledge the University of Kansas Computation Center for providing the necessary computer facilities for this project.

PRECEDING PAGE BLANK NOT FILMED.

ABSTRACT

A digital approximation of wavefront propagation described by Blum is implemented on a computer to obtain a medial point transformation of two-dimensional data. The transformation is found to be greated affected by noise and therefore unsuitable for pattern recognition. The transformation is shown to be useful for data compression and noise reduction. An aerial photograph taken over Phoenix, Arizona is processed using the transformation.

PRECEDING PAGE BLANK NOT FILMED.

TABLE OF CONTENTS

I. INTRODUCTION

1.1 Background

Recent developments in the remote sensing field have enabled investigators to sample and record vast amounts of video data. The great quantity of obtained data requires that methods be developed to process the data automatically. This processing generally takes two forms. First, the data may be processed to establish its content relative to the desired information; this establishes characteristics concerning shape, position and composition of objects included in the data and is termed pattern recognition. Second, the data may be processed to reduce its quantity by eliminating redundancy, so that the data may be efficiently transmitted or stored. This is the process of data reduction.

Harry Blum[1] presents the idea of classifying a pattern by considering its surface area of arc length. This surface area becomes a function of time by considering it as a normally propagating wavefront. Blum adds the restriction that when wavefronts meet, only the region of space through which waves have not passed are allowed to have waves generated in it. Figure 1.1 shows the wavefronts generated by two points for various values of time. Figure 1.2 shows the wavefronts obtained from a circle. Blum shows that the arc length as a function of time describes the geometry of the wave generating object.

A mathematical model of Blum's method of wave propagation was described by Kotelly[2] and implemented on a computer by Philbrick[3]. Both Kotelly and Philbrick suggested that the medial axis transformation resulting from such a wavefront propagation can be used for shape recognition. However, investigation by the author indicated that introduction of a small amount of noise on the image being transformed results in a profound change on the corresponding medial axis transformation. For this reason, it is believed that the medial axis transformation is not useful for shape recognition. Since Blum's method of wave propagation involves a transformation of two-dimensional data, however, it is believed that the method could provide a useful means of data compaction. The usefulness of wave propagation for data compaction was studied and the results are presented in this paper.

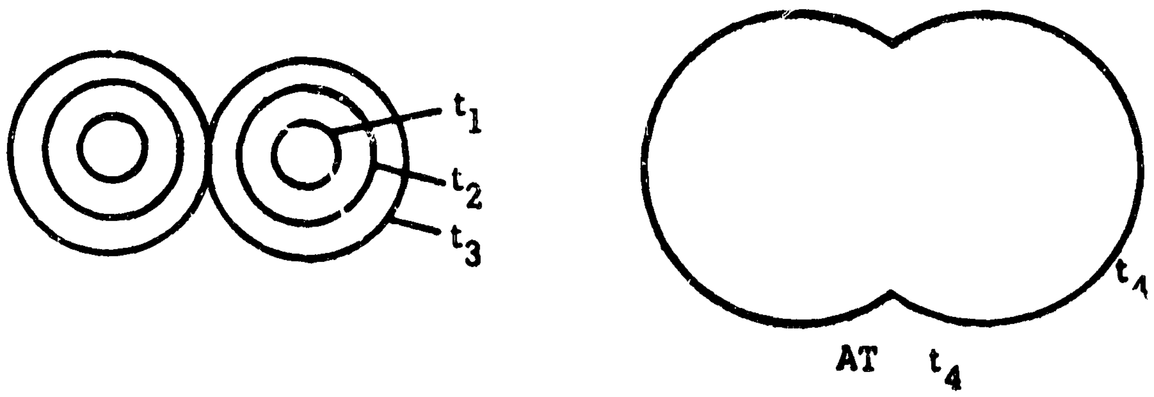


Figure 1.1

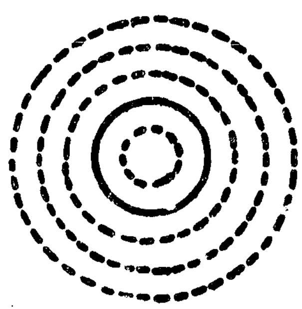


Figure 1.2

1.2 Digital Wavefront Propagation

Blum's concept of a propagating wavefront suggests that the geometrical properties of a two-dimensional figure can be transformed into a different and perhaps more compact form. In fact, a receding wavefront directly suggests a "shrinking" of data. However, to implement this method of data reduction on discrete data (such as would be used by a digital computer), it would be best to modify the concept of wave motion so that propagation is in vertical and horizontal directions, rather than in a normal direction, to take advantage of two-dimensional arrays in operating on images.

A computer simulation of the wavefront propagation was made using a GE 625 computer. The figure to be investigated was first assumed to be a solid, homogeneous image. This image was then superimposed on a grid of squares and each square was quantized according to whether the majority of the area of the square was inside or outside of the boundary of the figure. Those squares quantized inside of the boundary of the figure were assigned a value of one and those outside of the boundary were assigned a value of zero. These assigned values were then punched onto IBM cards and read into the computer as a two-dimensional array which shall be referred to as the object matrix.

The quantization of a circle is shown in Figure 1.3. The circle in Figure 1.3 (a) is quantized and put on IBM cards as shown in Figure 1.3 (b). Its appearance is enhanced in Figure 1.3 (c) by replacing the zeros by blanks and the ones by "X's." The computer outputs in figures 1.3 (b) and 1.3 (c) are distorted because a computer line printer prints ten characters per inch horizontally while it only prints six lines per inch vertically. Thus all computer outputs appear elongated in the vertical direction.

The computer simulation of wave propagation is accomplished as follows. Each point of value zero on the object matrix is assumed to propagate one unit in all vertical and horizontal directions. Thus, for an object matrix, A, of size NxN:

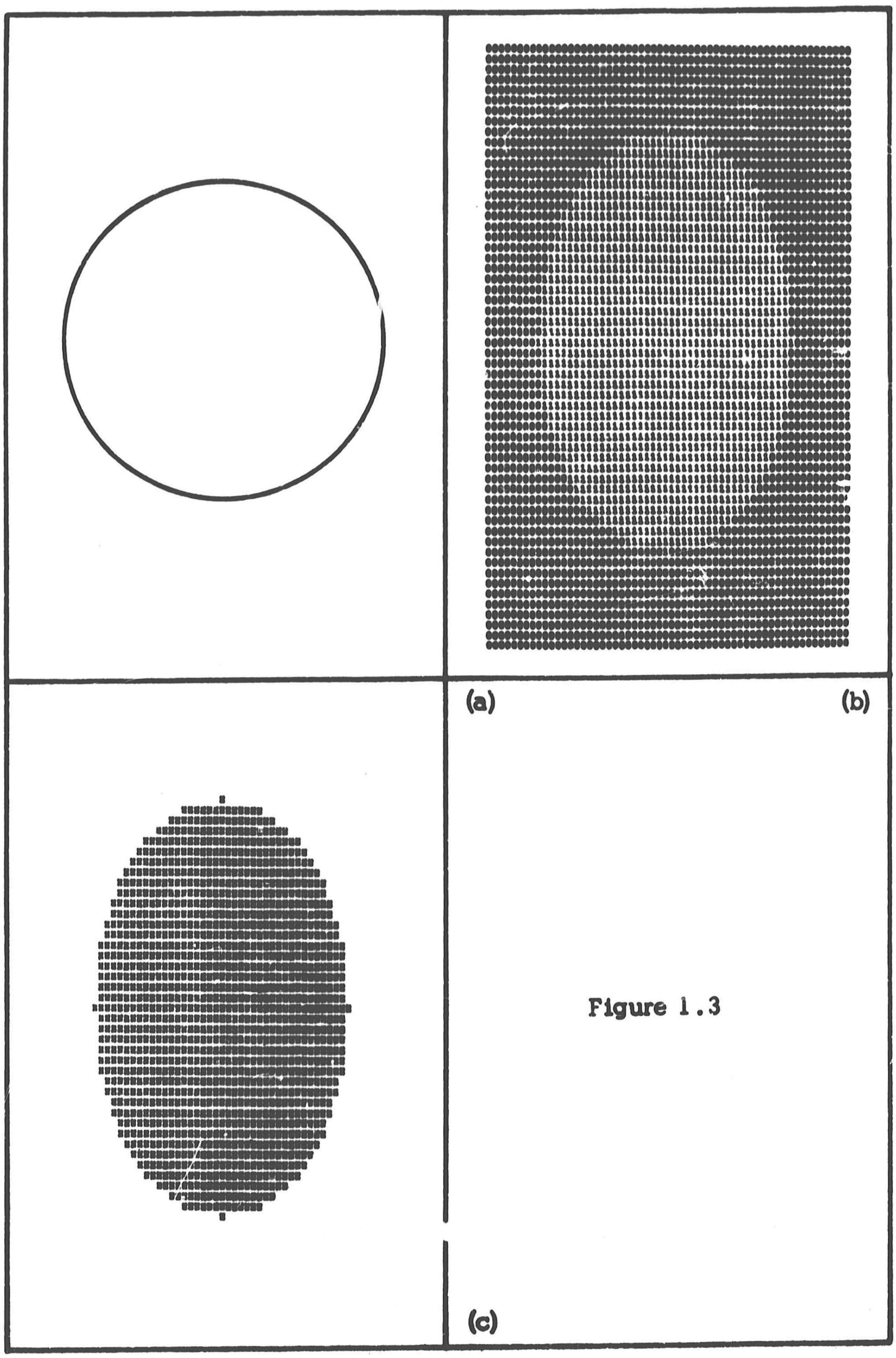


Figure 1.3

$$A_{i,j} = 0 \quad \Rightarrow \quad \left\{ \begin{array}{l} A_{i,j \pm 1} = 0 \\ A_{i \pm 1, j} = 0 \end{array} \right\} \text{ for all } i,j \quad 2 \leq i,j \leq N-1$$

Such an operation on the object matrix comprises one propagation iteration. Obviously, if a point adjacent to a propagating zero has a value of zero before an iteration, it will retain its value after the iteration. This process is repeated for each iteration of the propagation. Since each wavefront progresses by a matrix unit during each iteration, the iterations can be used as a measure corresponding to time in Blum's wave propagation.

To make a record of wave propagation, the points where the wavefronts intersect are marked. This is done with another matrix of size NxN which will be referred to as the skeleton matrix and which is initially set equal to zero. After each propagation iteration, each object matrix point which was changed to zero in the completed iteration which now has no immediately adjacent points of value one is labeled as a skeleton matrix point. It is labeled by setting the corresponding point on the skeleton matrix equal to a value equal to the number of the iteration just completed. In this matter, after all iterations are completed, the value of each non-zero point on the skeleton matrix represents the "time" at which an intersection of waves occurred at the corresponding point on the object matrix during the image shrinking process. Thus each skeleton point represents a medial point found in the shrinking process.

An example of the wave shrinking of a rectangle is shown in Figure 1.4. The object matrix is shown in Figure 1.4 (a). Part of the skeleton formation has taken place in Figure 1.4 (b) where the "X's" represent the points of value one remaining on the object matrix while the numbers represent the non-zero values of the corresponding points on the skeleton matrix. 1.4 (c) shows the skeleton matrix after all iterations have been completed. (Henceforth, in any reference to a skeleton matrix, it will be assumed that all propagation iterations have been completed. References to an object matrix will assume that no propagation

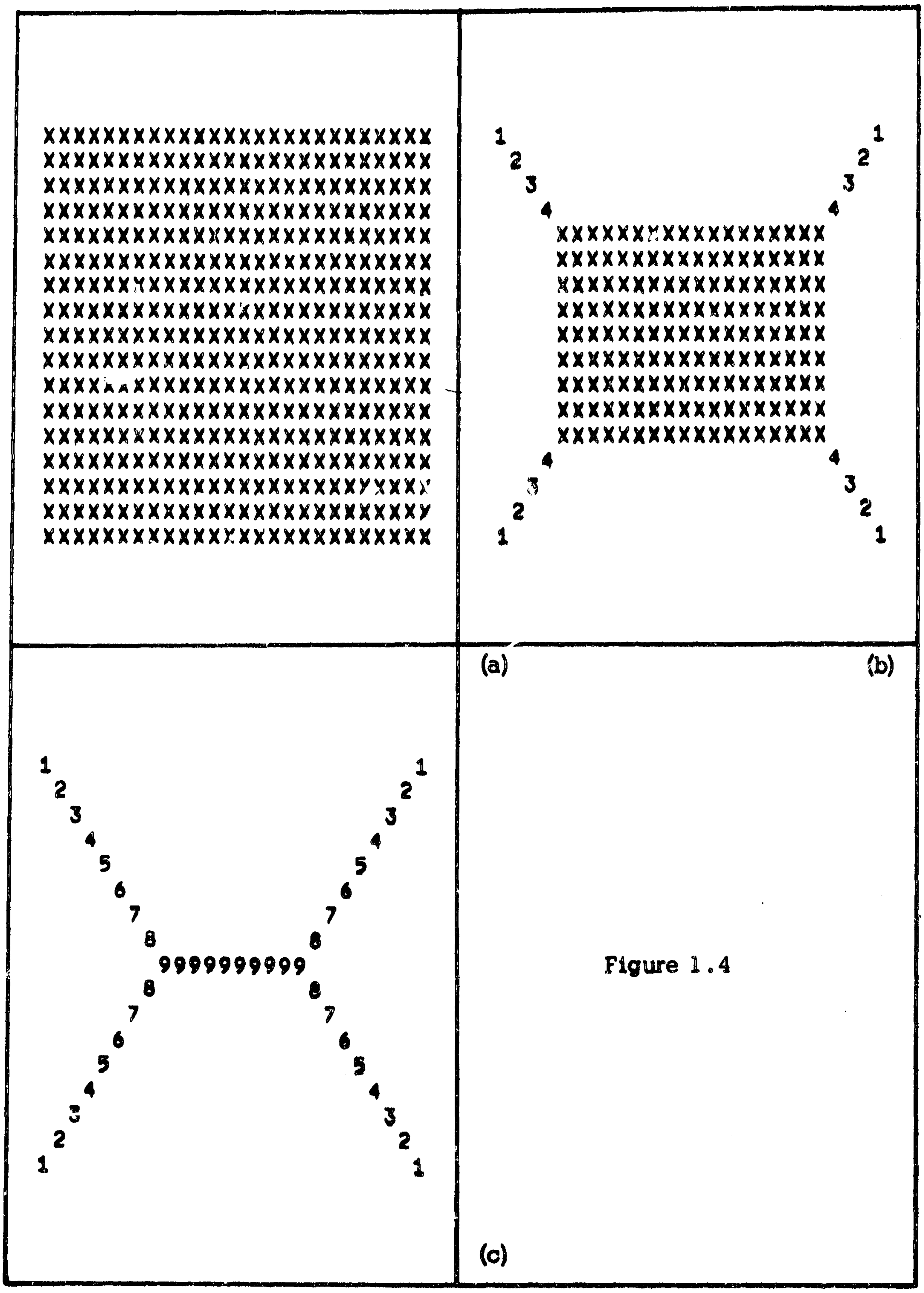


Figure 1.4

iterations have been made.) The skeleton matrix for the circle in Figure 1.3 is shown in Figure 1.5 where the non-zero values are shown as stars.

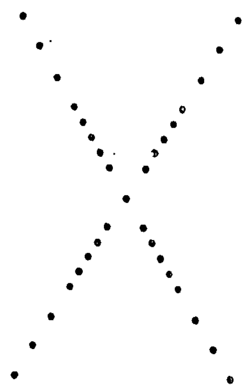


Figure 1.5 Skeleton Matrix for the Circle in Figure 1.3

1.3 Objectives

A skeleton matrix uniquely describes a corresponding object matrix. Thus the two matrices are transforms of each other and, in fact, the object matrix can be simply obtained from its corresponding skeleton matrix as will be shown in section 2.1. This transformation can be useful in at least two areas of remote sensing, and these two areas will be investigated in the remainder of the paper.

The first purpose in investigating this method of data transformation is to determine its usefulness in data compaction. Data compaction is an extremely important problem when two-dimensional data is to be transmitted from one place to another such as with the telemetry of video data from a spacecraft to another receiving station. Also, data compaction is important in various kinds of information retrieval problems. In describing geometric areas by arrays of points, it might be useful to consider how orientation of both the image and the object array points affect the data compaction. Also, it would be useful to know which skeleton array points are most useful in describing the corresponding object matrix.

The second objective in studying this transformation is to determine its usefulness in noise reduction.

Two considerations are made in the investigation. First, the methods described must be easily adaptable to computer programming or logic design methods. Second, some application of the evolved methods should be made to actual photographic data. It is felt that application of these considerations provides a quick measure of the usefulness of the transformation.

II. ORIENTATION

2.1 Description by Diamond Fitting

It has been shown how a two-dimensional figure may be described by several points with numerical values on a skeleton matrix. The values of these points represent the number of the iteration in the shrinking process when the corresponding point on the object matrix becomes isolated from other points of value one and is itself assigned a value of zero. Obviously, then, the number of iterations in the entire shrinking process corresponds to the largest value assigned to a member of the skeleton matrix since the last shrinking reduction of the object matrix leaves no points of value one to be neighbors to the points which receive a value of zero in the last shrinking reduction.

The process of object matrix shrinking can be reversed to obtain an object matrix from its corresponding skeleton matrix. This is done on a matrix which is initially set equal to zero. The first step in reforming an object matrix is to consider the point or points with the largest value on the skeleton matrix. The corresponding points on the reforming object matrix are set equal to one. This corresponds to a reversal of the last iteration in the object matrix shrinking process. If the largest value on the skeleton matrix is assigned a value of N_{LARGE}, the reformation process can continue by next considering all points on the skeleton matrix of value $N_{LARGE-1}$. In this step, all points on the reforming object matrix which are horizontal or vertical neighbors to points of value one are first themselves assigned a value of one, and then points on the reforming object matrix corresponding to points of value $N_{LARGE-1}$ on the skeleton matrix are assigned a value of one. The same process is repeated for the second step except that skeleton matrix points of value $N_{LARGE-2}$ are considered.

Each succeeding step considers a value one less than was considered in the previous step. When skeleton matrix points of value one have been considered, the object matrix has been reconstructed.

Each point of value one placed on the object matrix by the skeleton matrix in the reformation process is allowed to propagate to its neighbors in the next step and the resulting points are allowed to propagate to their neighbors in the following step. If a single point were placed on a matrix and allowed to propagate in such a manner it would take on the appearance of a diamond like the one shown in Figure 2.1 and would grow with each step of the propagation. Since the skeleton matrix is made up of several points, the object matrix may be considered as the union of several diamonds, each described by a point on the skeleton matrix. If a point on the skeleton matrix has a value of one, the corresponding diamond will only be a single point. For points on the skeleton matrix of larger values, the corresponding diamond will have a number of points on each of its sides equal to the value of the corresponding skeleton matrix point.

Figure 2.2 (a) shows an object matrix representing a small circle. Figure 2.2 (b) shows the corresponding skeleton matrix. By drawing diamonds centered at the points indicated with sizes indicated by the value of the skeleton matrix, the original figure can be reconstructed. Several of these diamonds are shown on the object matrix in Figure 2.2 (c).

It has been shown that the skeleton matrix transformation may be regarded as a description of a figure by diamond fitting. The position of each diamond is described by the position of the corresponding point on the skeleton matrix, and the size of the diamond is described by the value of the corresponding skeleton point. Such a method of description becomes convenient when resolution is considered. Suppose, for example, that in the reconstruction of a figure, it is not desired to consider detail which is less than three units on a side. To do this, it is only necessary to ignore the points on the skeleton matrix with a value less than three in reconstruction. Such an approximation has been made on the group of rectangles shown in Figure 2.3 (a). The result is shown in Figure 2.3 (b). Because the points on a skeleton matrix can be directly used for resolution determination, the values of these points will be referred to as resolution values.

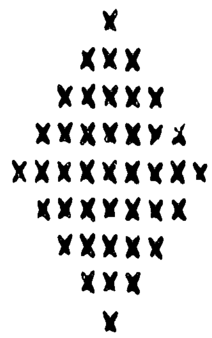
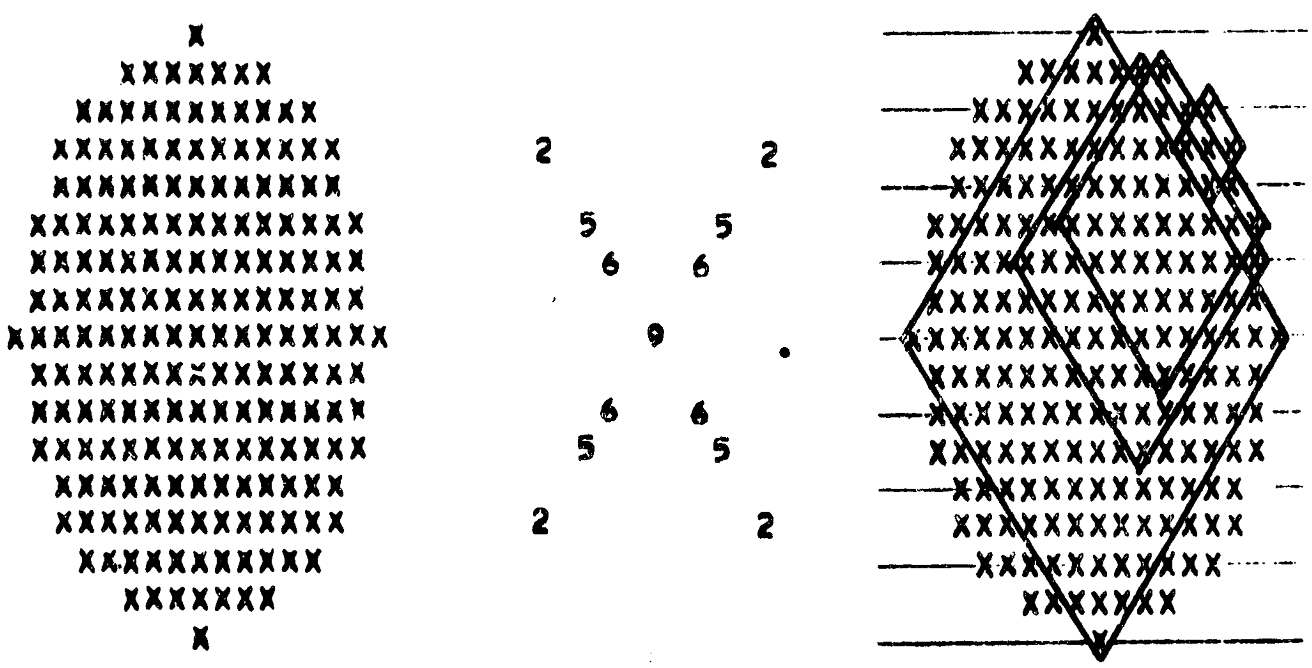


Figure 2.1



(a) (b) (c)

Figure 2.2

The method of diamond fitting can be applied to several aspects of pattern description. Transmission of video data can be controlled by eliminating skeleton points of low resolution. Such a method is useful when it is not necessary to transmit the detail of all of the data collected. Diamond fitting can be used to reduce noise. Such a use will be discussed in Chapter 3. Diamond fitting can also be used in pattern recognition to describe geometries and orientations of patterns.

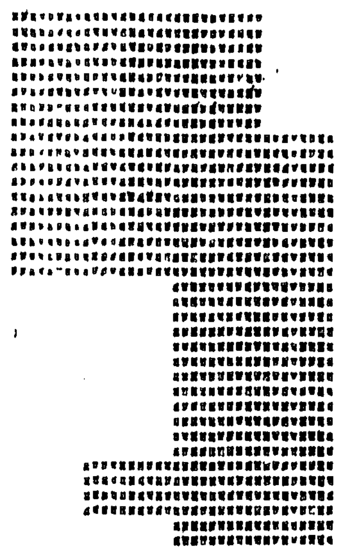
2.2 Preferred Orientation

Since images are being described by diamond fitting, the geometry of a diamond would suggest that it would best orient the figures so that the majority of the straight edges are parallel with the edges of the diamonds which represent the figures. Thus, it would be expected to be preferable to orient figures at an angle of 45° from a vertical direction.

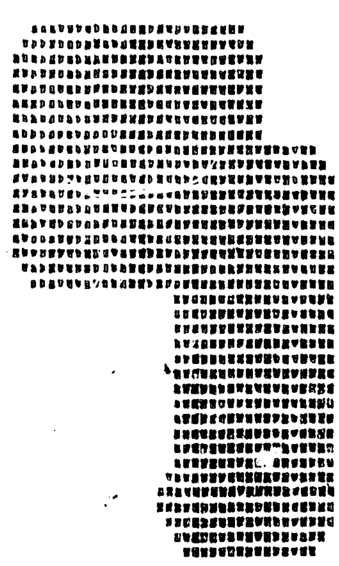
To study the effects of the orientation of a figure, a top-view outline of a BAC-145 British trainer jet was used. This outline is shown in Figure 2.4 (a). The outline was oriented at 0°, 15°, 30°, and 45° from the vertical direction and coded onto IBM cards. The quantized outlines for the jet are shown in Figures 2.4 (b), (c), (d), and (e) respectively. As the angle of the orientation increases, the number of 45° lines in the quantized outline increases and the number of vertical and horizontal lines decreases. Thus, the effects of diagonal orientation become more pronounced as the angle of the jet outline orientation increases. This would provide the best data for describing the figure since a large amount of the edge of the diamonds would coincide with the edge of the figure. Thus the diagonal orientation would be the "preferred" orientation.

2.3 Measurement of Orientation

Several properties of the diamonds in the figures representing the jet were calculated for various orientations. The properties calculated



(a)



(b)

Figure 2.3

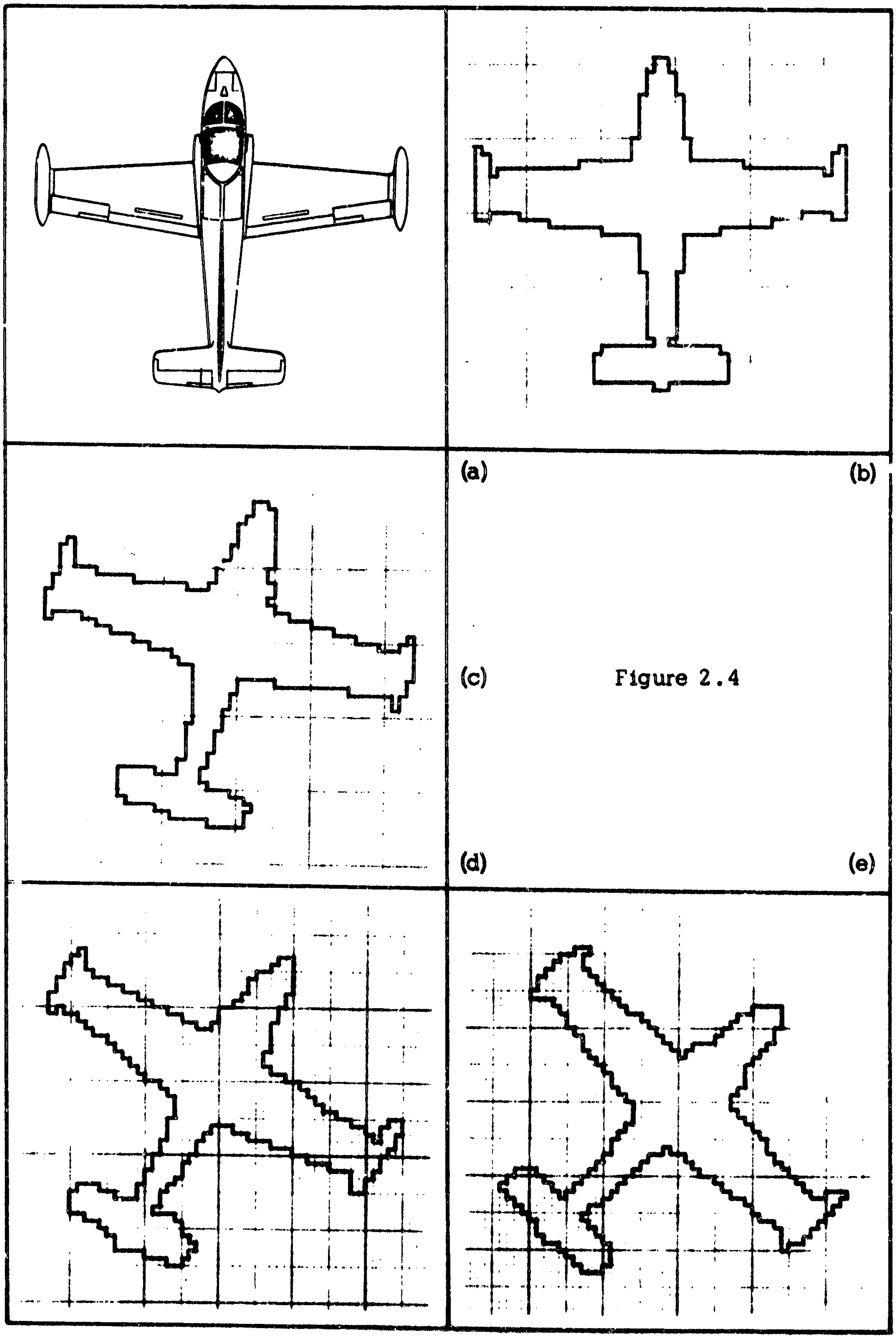


Figure 2.4

were: the average resolution of the diamonds in the figure; the average number of points in each diamond which lies on the boundary of the figure; and the number of diamonds in the figure. The results are shown graphically in Figure 2.5.

The results show that the average resolution per diamond increases as diagonal orientation of the jet is approached. The average boundary surface area per diamond also increases with the orientation angle while the number of diamonds decreases. The largest variation is in the average boundary surface area per diamond indicating that the boundary surface property is probably the best measure of image orientation. The results also show that with an orientation of 45°, the figure can be described with fewer data points than with other orientations and that the average data point describes a diamond with more area than data points of the same figure with other orientations. This means that with the use of the skeleton matrix transformation which has been described and with orientation of the figure so that straight edges are at an angle of 45° with the vertical, more area can be described with fewer data points than can be described with other orientations. This, of course, is desirable.

2.4 Advantages of Square Fitting

An orientation of 45° provides a rather awkward angle with which to work. A person is not accustomed to orienting the straight edges of figures at angles of 45°, but rather in vertical and horizontal directions. It is easier to visually judge proper vertical and horizontal orientation of a line than it is to judge proper 45° orientation of a line. Thus, one feels more at ease with photographic imagery in which the straight lines in it have been oriented in directions parallel to the edge of the photograph rather than at other angles such as 45°. Orientations of 45° are also disadvantageous when the representation of the image is elongated, such as the computer outputs in this paper, or otherwise linearly distorted in vertical or horizontal directions. Apparent diagonal angles will

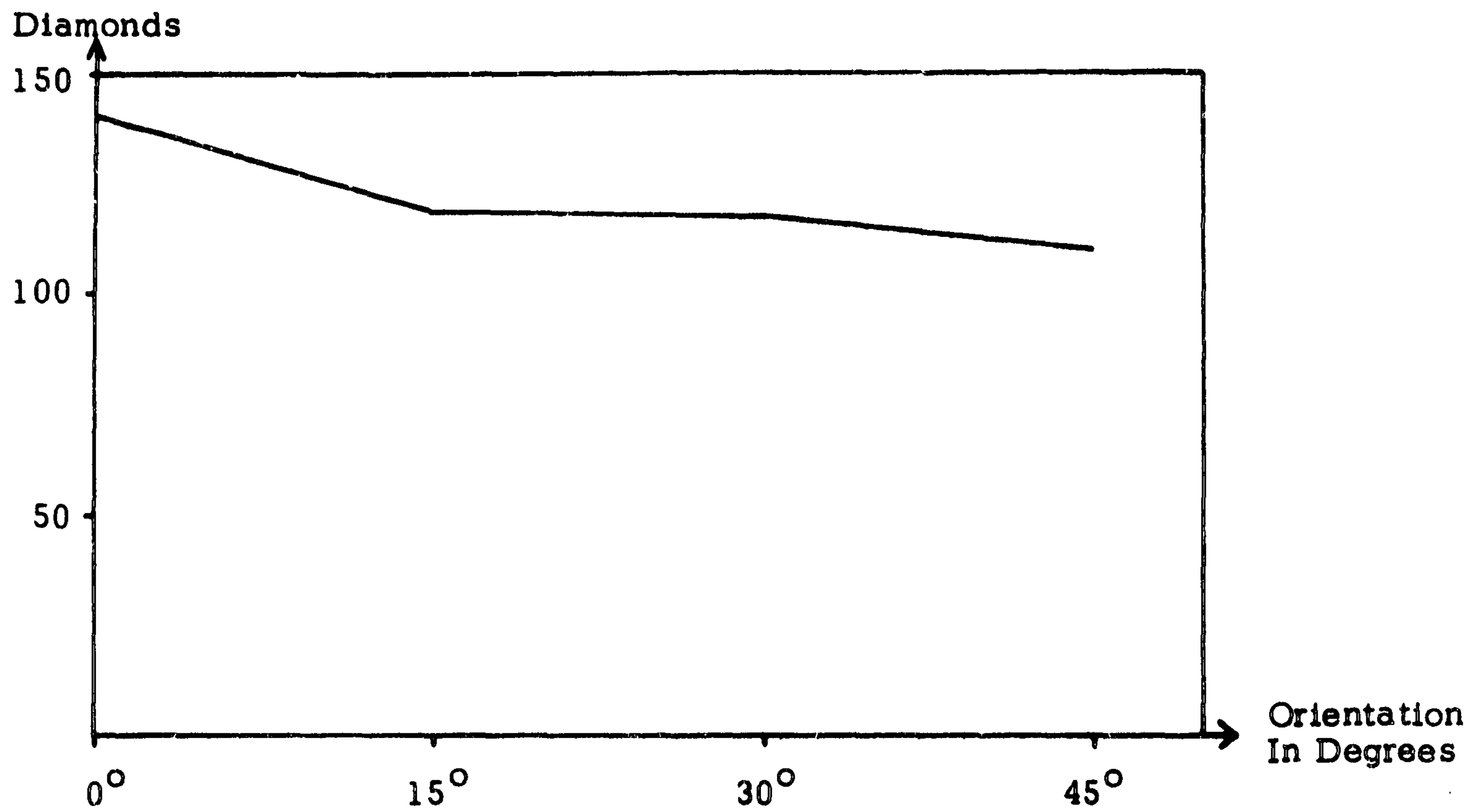


Figure 2.5(a). Number of Diamonds in Figure Vs. Orientation For BAC-145.

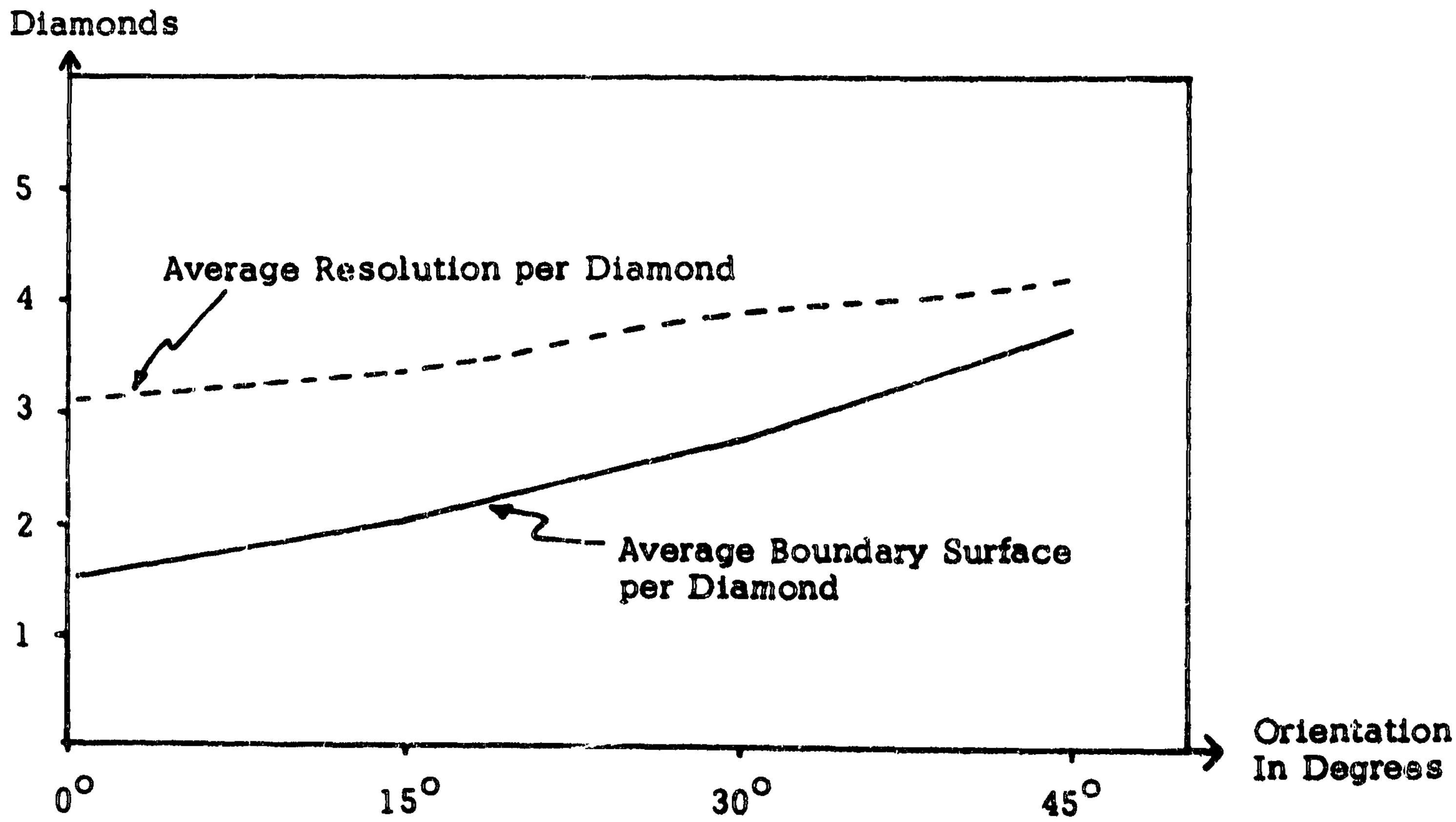


Figure 2.5(b). Average Resolution Per Diamond and Average Boundary Surface Per Diamond Vs. Orientation For BAC-145.

change under such conditions although orientations of vertical and horizontal lines will not.

The difficulties encountered with use of a 45° orientation would suggest that a method of image description using square fitting rather than diamond fitting would be more appropriate for use. Such a method would allow images which contain rectangular forms to be represented more easily and naturally. In fitting squares into rectangles, the edges of the squares would often coincide with the edges of the rectangles. Thus, the edges of the rectangular forms would tend to be better enhanced with a square fitting transformation than with a diamond fitting transformation.

To develop the method of image transformation by diamond fitting, a matrix of numbers with the numbers in rows and columns was used. An obvious way to obtain an array for diamond fitting would be to rotate our matrix by 45°. With the resulting arrangement of points in the array, the four nearest neighbors to each point would be in diagonal directions from that point rather than in vertical and horizontal directions An array of points resulting from such a rotation is shown in Figure 2.6

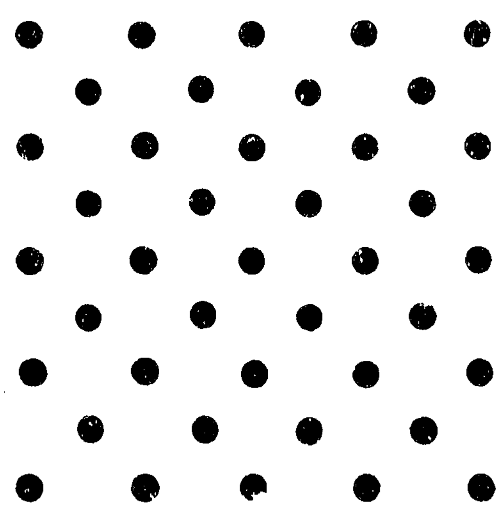


Figure 2.6 An Array of Points
for a Square Fitting Transformation

2.5 Translation from Diamond Fitting

To make most effective use of the method of image description using the square fitting transformation, it would be best to record image data for the object array at points in staggered rows. Such a method of recording data would provide a direct correspondence between points on the object array and the corresponding areas of the image. However, image data is usually not recorded in this manner, but, instead, it is usually quantized in straight rows and columns. For this reason, it would be desirable to devise a method to translate data in the object matrix, which is in rows and columns, to data in staggered rows which can then be described using the square fitting transformation.

A simple way of translating the data in the object matrix is to rotate the object matrix 45°. However, square fitting after such a rotation would only amount to performing the same operations which would be made if diamond fitting were done before the rotation. Thus the advantages to be gained from square fitting which have been discussed would not be gained from such a translation. Furthermore, storage of such an array in a computer would be awkward since computers store rectangular arrays and the rotated matrix would be a diamond shaped array. A better method of object matrix data translation is desirable.

The method of data translation used in this paper involves matching of two matrices. First, the quantized data is read into a square matrix as was done for the diamond fitting transformation. Then each point of a second matrix which is of the same size is set equal to the corresponding data point on the first matrix. This results in two matrices which are initially made identical to describe the object array. Each point in the second matrix is assumed to be down and to the right of the corresponding point of the first matrix such that each interior point of the object array has four identically close neighbors, each located in diagonal directions from that point The formation of such a staggered row array is shown in Figure 2.7. The solid dots represent points on the first matrix while the hollow dots represent points on the second matrix. The arrows indicate which points on the respective matrices are matched.

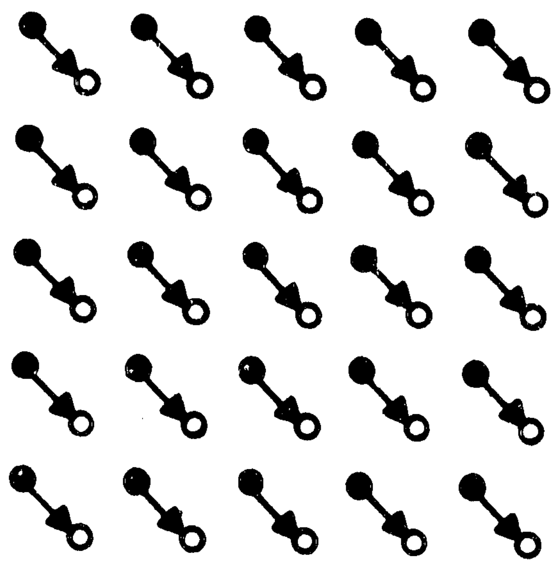


Figure 2.7

A better method of obtaining an array for square fitting from an array for diamond fitting might be to define the initial value of each point on the second matrix according to all four of its neighbors rather than by just one neighbor as is done in the translation method just described. However, the much simpler method of point to point matching of values with one neighbor is used in favor of this more sophisticated method which would involve a decision process for each point on the second matrix.

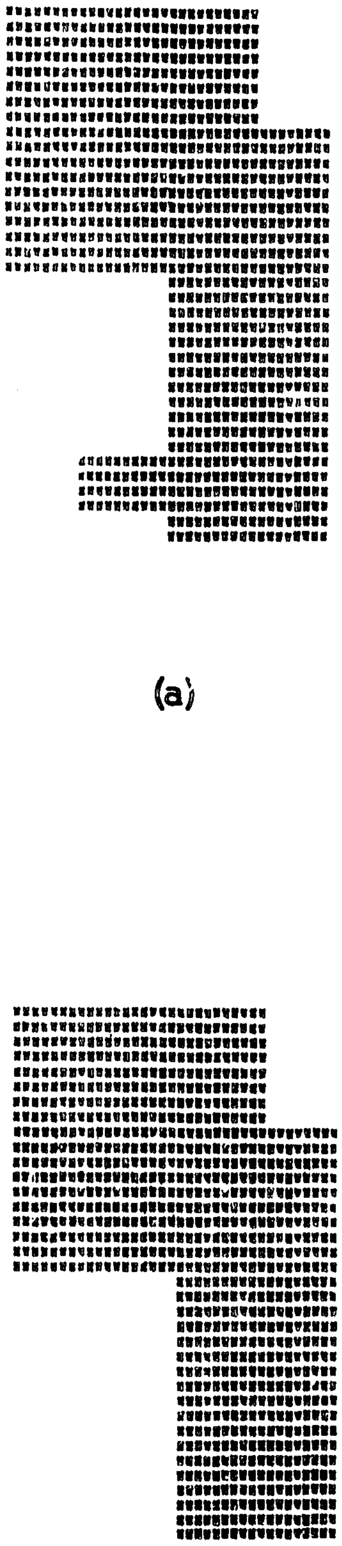
Of course, another method of obtaining a square fitting array would be to shift every other row in the object matrix horizontally by one-half of a unit. This method, it is felt, would limit the resolving power of the square fitting method of pattern description and is therefore not used.

The method of conversion using matrix matching, then, is used in preference to the other methods mentioned for several reasons. It is quick and simple. The effect on the processing of the image would be the same for both the vertical and horizontal directions. It allows the resulting skeleton array to maintain close control on the resolution during the object array reformation process.

2.6 Examples of Square Fitting

The advantages of describing an image by square fitting can be shown by considering the set of rectangles shown in Figure 2.8 (a). These rectangles are identical to the rectangles shown in Figure 2.3 (a). However, this figure can be completely reconstructed by considering points from the skeleton array which describe squares with resolutions greater than three. This means that the skeleton matrix contains no points with resolution less than four. In fact, a reasonable approximation to the figure can be made by considering squares described by the skeleton array which have resolutions equal to or greater than eighteen. Such an approximation is shown in Figure 2.8 (b). Also, the skeleton for the object matrix in Figure 2.8 (a) obtained using the square fitting transformation has fewer points than the skeleton obtained using the diamond fitting transformation. Thus, since the given object matrix can be described with a skeleton matrix containing fewer points with higher resolution values by using square fitting rather than diamond fitting, it would appear that the method of square fitting is best suited for rectangular images.

The effect of orientation on the square fitting transformation was checked using the same data of the BAC-145 trainer jet which was used in the first part of this chapter. The results for the average resolution of the squares and the average surface boundary for the squares are graphed in Figure 2.9. The average resolution and the average surface boundary are greatest when the figure is oriented at an angle of 0° from the vertical. This would indicate that the preferred orientation using the square fitting transformation is in a vertical or horizontal direction.



(a)

(b)

Figure 2.8

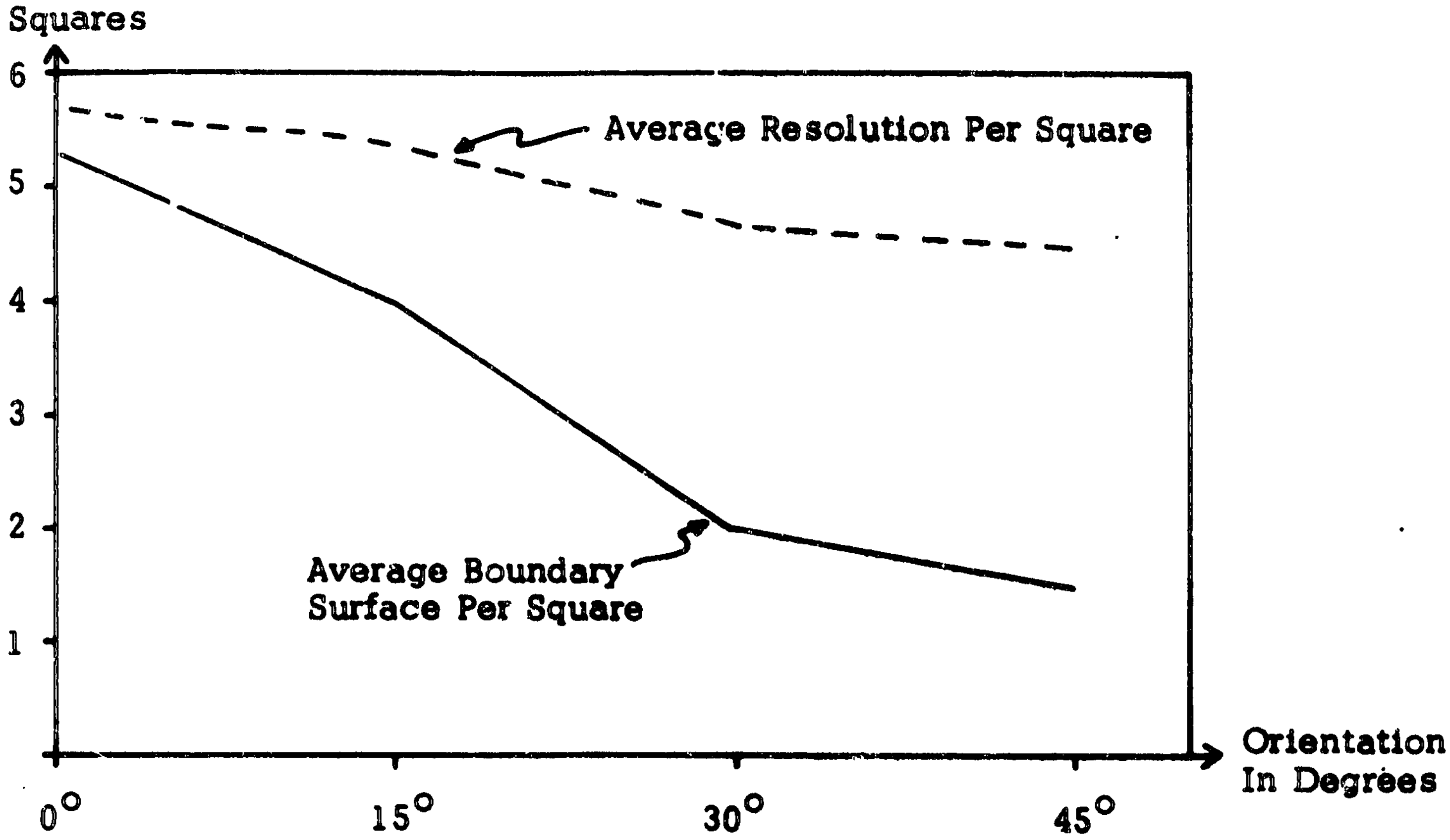


Figure 2.9 Average Resolution Per Square and Average Boundary Surface per Square Vs. Orientation for BAC-145.

Such an orientation is desirable for the reasons outlined in section 2.4.

2.7 Weight

The resolution of both a diamond and a square has been defined as the number of points on each edge of that figure. Such a definition is useful since resolutions are obtained directly in the skeleton transformation process. Another useful property to consider would be the number of points which are included in each diamond or square. Obviously, such a property would provide a measure of the "area" included in each diamond or square. This property will be termed as the weight.

The weight of a diamond or square provides a direct measure of the diamond's or square's usefulness in the formation of an object matrix. A primary objective in two-dimensional data reduction is to describe large areas with as few data points as possible. Since weight is a measure of area, diamonds or squares of large weights provide the most economical means of describing a figure. Diamonds or squares of small weight contribute little to the description of an area, and, if fine resolution is not required, can often be ignored. In such cases, the small diamonds or squares are considered noise and should be eliminated for economical description of a figure.

The weight of a diamond or square can be easily obtained from its resolution. In the case of a square, the points in the square consist of two square matrices, each with the same center. The dimension of the larger matrix is equal to the resolution of the square which it describes. The dimension of the smaller matrix is equal to the resolution of the larger matrix minus one. Summing the number of points in each matrix together, the weight of a square becomes:

$$W = R^2 + (R - 1)^2$$

where W is the weight of the square and R is its resolution. The formula also holds for the case of a diamond.

As an example, consider the calculation of the weight of the diamond in Figure 2.1. The resolution, R, of the diamond is five. Its weight can be calculated using the formula

$$W = 5^2 + (5 - 1)^2$$
$$= 41$$

This value of W gives the number of points in the diamond.

III. NOISE REDUCTION

3.1 Edge Enhancement

Chapter two discussed the orientation of images. It was concluded that for images with vertical and horizontal edges, the square fitting transformation provides a better method of data compression than does the diamond fitting transformation. This is true because the edges of the images which are being described tend to coincide with the edges of the squares. Such a tendency for edge coincidence is used to "straighten out" rough vertical and horizontal edges and act as an edge enhancer.

Enhancement of edges is useful in improving the visual appearance of an image. Straight edges appear more pronounced and regions formed by the boundaries appear more distinct.

Enhancement of edges is also useful since it simplifies the skeleton resulting from a square fitting transformation of an image. If the edges are enhanced, then several skeleton points with resolutions of small values are eliminated, since only large squares are used to describe the image by fitting and small squares are not needed to fill in the once-jagged edges of the image. A simplification of the skeleton with a negligible loss of information is desirable since the skeleton is used as the means of data reduction.

Besides providing edge enhancing in vertical and horizontal directions, the square fitting transformation is also useful in general noise reduction. Noise on two-dimensional imagery generally appears as isolated single points or small groups of points. When a skeleton transformation is made of the points which describe the noise, the corresponding skeleton matrix points have values of small resolution. The points on the object matrix corresponding to these skeleton points can then be identified and considered as noise, and steps can be taken to change their object matrix value.

3.2 Implementation

In eliminating noise from a square matrix of discrete points, the most logical way to begin is to find all points on the object matrix whose neighbors have a different value and change the value of each of these points to the value of their neighbors. For example, if a point on the object matrix, $A_{i,j}$, has a value of one, and $A_{i,j+1}$, $A_{i,j-1}$, $A_{i+1,j}$, $A_{i-1,j}$ each have a value of zero, then the point $A_{i,j}$ would be considered as noise and assigned a value of zero. This process eliminates isolated points of noise in otherwise well defined regions and provides a substantial reduction of noise on noisy imagery.

Procedures for further enhancement and noise reduction of imagery are not so obvious as the step just described. It is necessary to establish a criterion for deciding which object matrix points should have their values changed and what their new values should be. To make use of the advantages afforded by the use of the square fitting skeleton transformation which have been described, this criterion should base its decision process on the skeleton transformation of the image which is being enhanced. This would suggest that the second step in the noise reduction process is to take the skeleton matrix transformation of the image.

At this point, the concept of the skeleton matrix transformation must be modified to allow the transformation of matrices with more than one type of region. If each region is represented by a numerical value in the object matrix, then the number of types of regions is the number of different values in the object matrix. For example, all of the object matrices which have been considered thus far could be regarded as images with two regions. One region consists of the area in which all of the object matrix points have a value of one and the other region consists of the area where all of the points have a value of zero. Finding the skeleton matrix transformation of the image, then, consists of taking the skeleton matrix transformation of each region separately and then considering them together. The skeletons for the different regions do not intersect since each skeleton is within the area occupied by its corresponding region in the object matrix. However, it is necessary to remember from which region each skeleton point was formed. Images with more than two types of regions will be considered in chapter five.

In the matrix which results from the noise reduction process, it is desirable to have each point assigned to a particular region. However, in the intermediate steps of the decision process of noise reduction, the points representing noise have an undetermined value. These points are called free points. The skeleton matrix is used to determine which points in the object matrix are free points. This is done by choosing the points on the object matrix which do not belong to diamonds, or in the square fitting case, squares which have at least a certain resolution value.

Once the free points have been determined, it is necessary to use a decision process to assign values to them. The reader can think of several ways to assign these values. For purposes of investigation and comparison, four simple methods are used here. Flow charts for these methods are shown in Appendix B.

<u>Method 1</u>

All points described only by skeleton points with a resolution value of one are free points. The value of each free point is assigned according to the values of its four neighbors. The value which occurs most frequently among its neighbors' values is the value assigned to the free point in consideration. If there is no single value which occurs most frequently, the free point is assigned its original value.

<u>Method 2</u>

All points described only by skeleton points with a resolution value of one are free points. The regions described by the non-free points are allowed to grow into the areas occupied by the free points. This process is done by assigning free points which are neighbors of one or more non-free points a value which is equal to that of one of the non-free neighbors. The newly assigned free points then become non-free points and the growth is repeated several times until all free points are eliminated.

Method 3

Method 3 is like method 2 except that the growth process is not repeated. Instead, free points remaining after the first step of the growth process are assigned their original value.

All three methods consider free points as points which correspond to skeleton points of resolution one. Also, each of the three methods may be reinforced again. That is, the matrix resulting from the enhancement process may be considered as an object matrix and the same process can then be applied to that matrix. This reinforcement may be used as many times as is desired.

Method 4

Method 4 is like method 3 except that with each reinforcement, the maximum size of the resolution values associated with each free point is increased by one with each reinforcement.

3.3 Examples and Results

To check the four methods of noise reduction, three figures were used. The first was the group of rectangles shown in Figure 2.3 (a), the second was the circle shown in Figure 1.3 (c), and the third was the BAC-145 at 0° shown in Figure 2.4 (b). A random number generator with a uniform distribution between zero and one was used to add noise to the images. A random number was generated for every point on each matrix. If the random number was greater than 0.8, the value of the matrix point was changed. The images were processed by the methods discussed for five reinforcements. The square fitting transformation was used.

The effectiveness of the noise reduction methods was checked by comparing the enhanced images with the original noiseless images. A percentage of recovery was calculated for each image. This value represents the percentage of points which were changed in the noise making process that were then changed back to their correct value in the enhancement process.

To provide a comparison between the methods discussed and more conventional noise reduction methods, the noisy images were also processed by assigning a value to each point according to the value of each of the majority of its neighbors. This process was also done for five reinforcements.

Some results from the group of rectangles are shown in Figures 3.1 and 3.2. Figure 3.1 (a) shows the rectangles before noise was added. Figure 3.1 (b) shows the rectangles after noise was added. Figure 3.1 (c) shows the noisy rectangles after they were processed using the conventional neighbor search method with three reinforcements. Figures 3.2 (a), (b), (c) and (d) show the results after three reinforcements for methods one, two, three, and four respectively.

Graphs of percentage recovery vs. number of reinforcements for the methods of noise reduction discussed are shown in Figures 3.3, 3.4, and 3.5. The results using the groups of rectangles are shown in Figure 3.3. The circle is considered in Figure 3.4 and the BAC-145 at 0° is used in Figure 3.5.

The results show that method four apparently provides the best noise reduction, although method two provides the best noise reduction if only one reinforcement is used. Of course, many other methods of noise reduction can be contrived using the skeleton transformation. The results given here show that methods using the skeleton matrix transformation can be made to be effectively useful in reducing video noise.

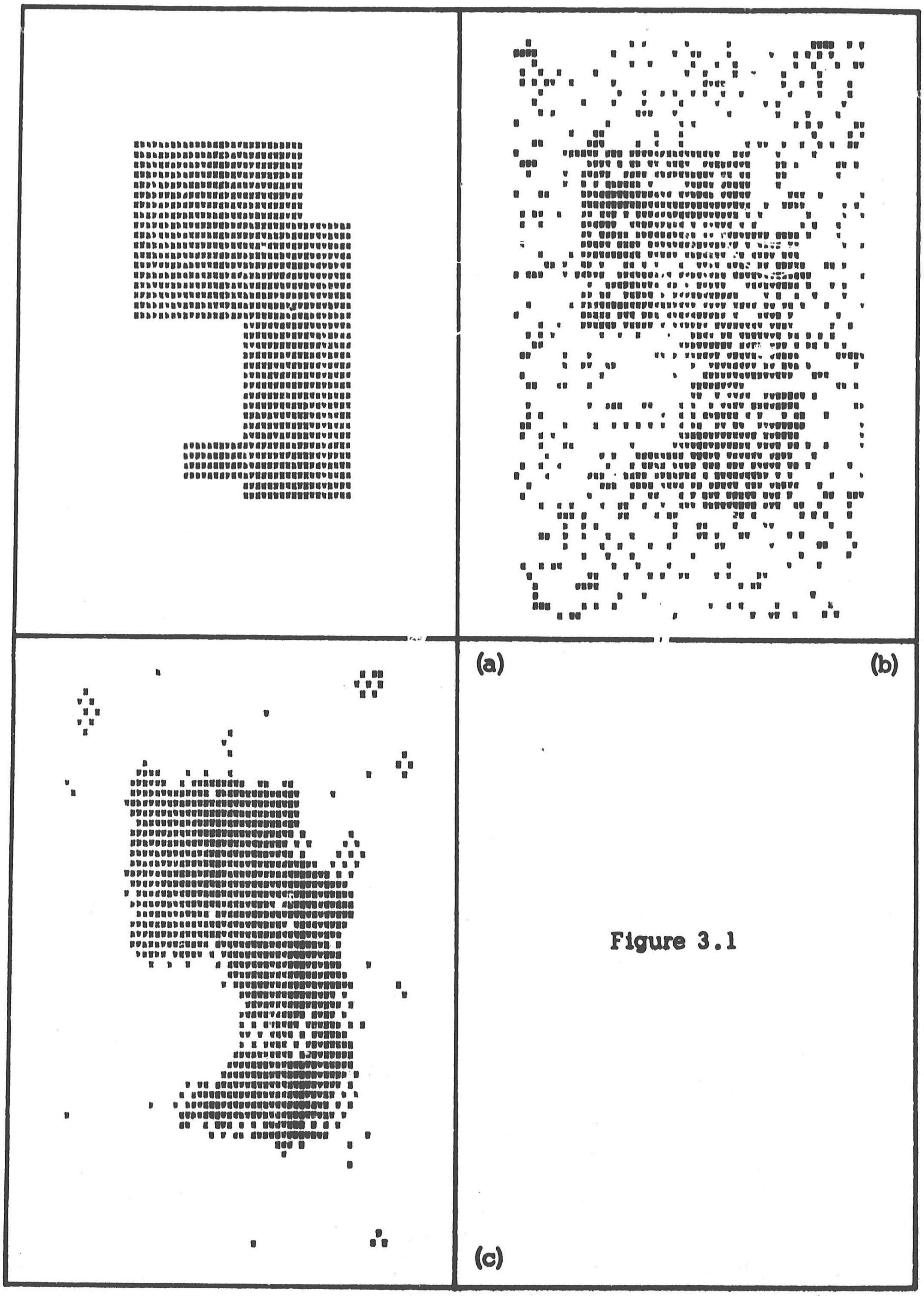


Figure 3.1

Figure 3.2

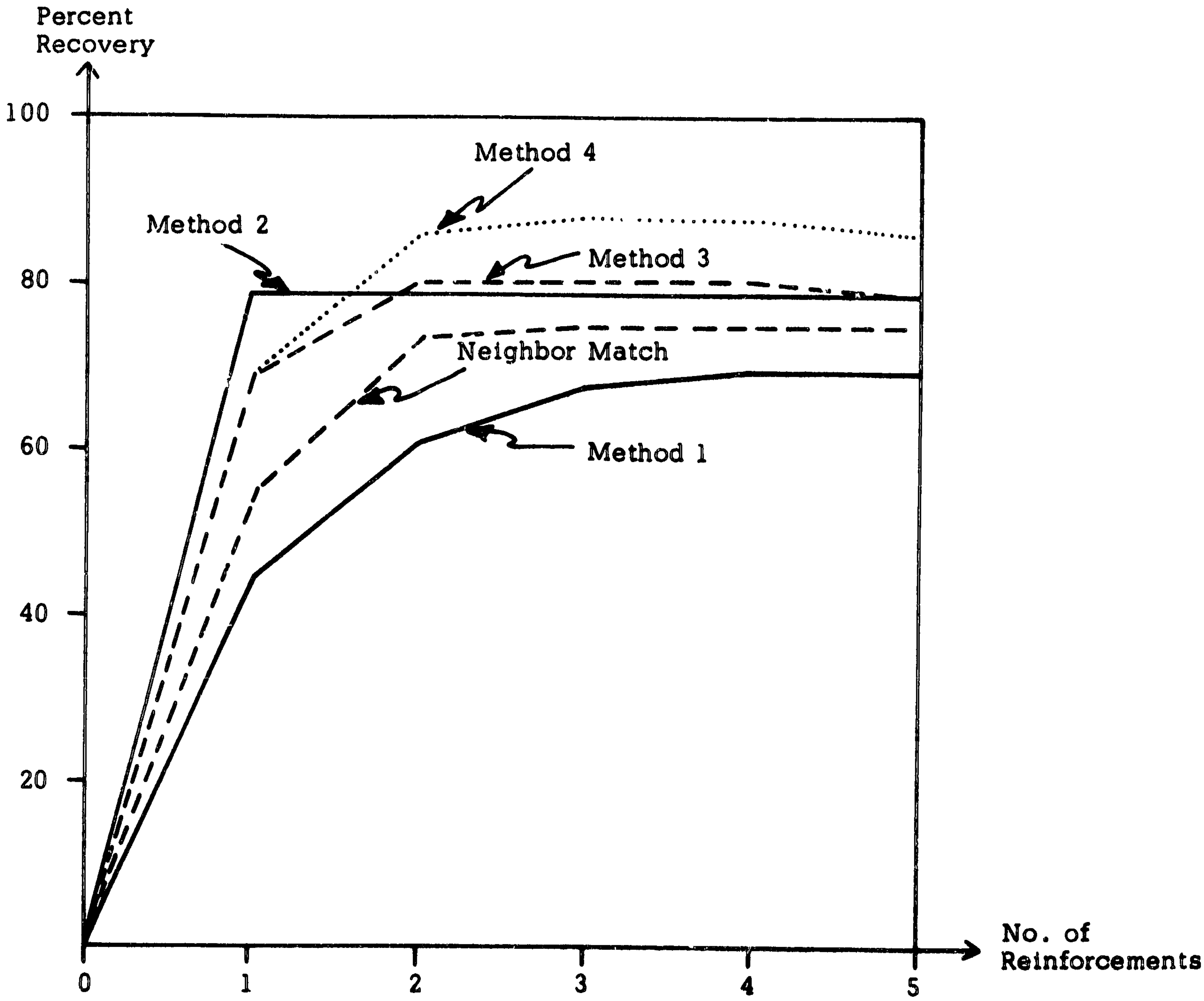


Figure 3.3. Rectangles

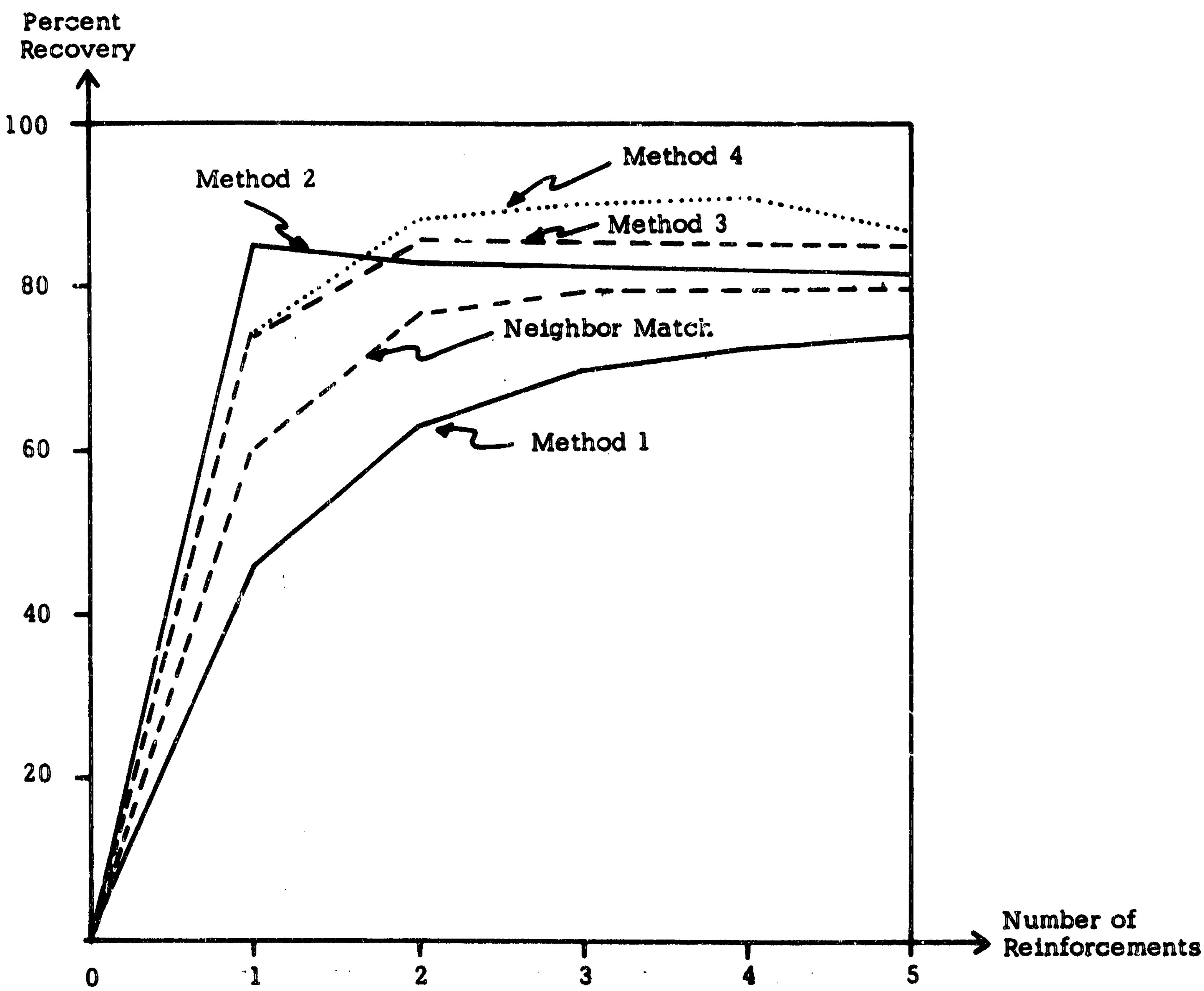


Figure 3.4. Circle

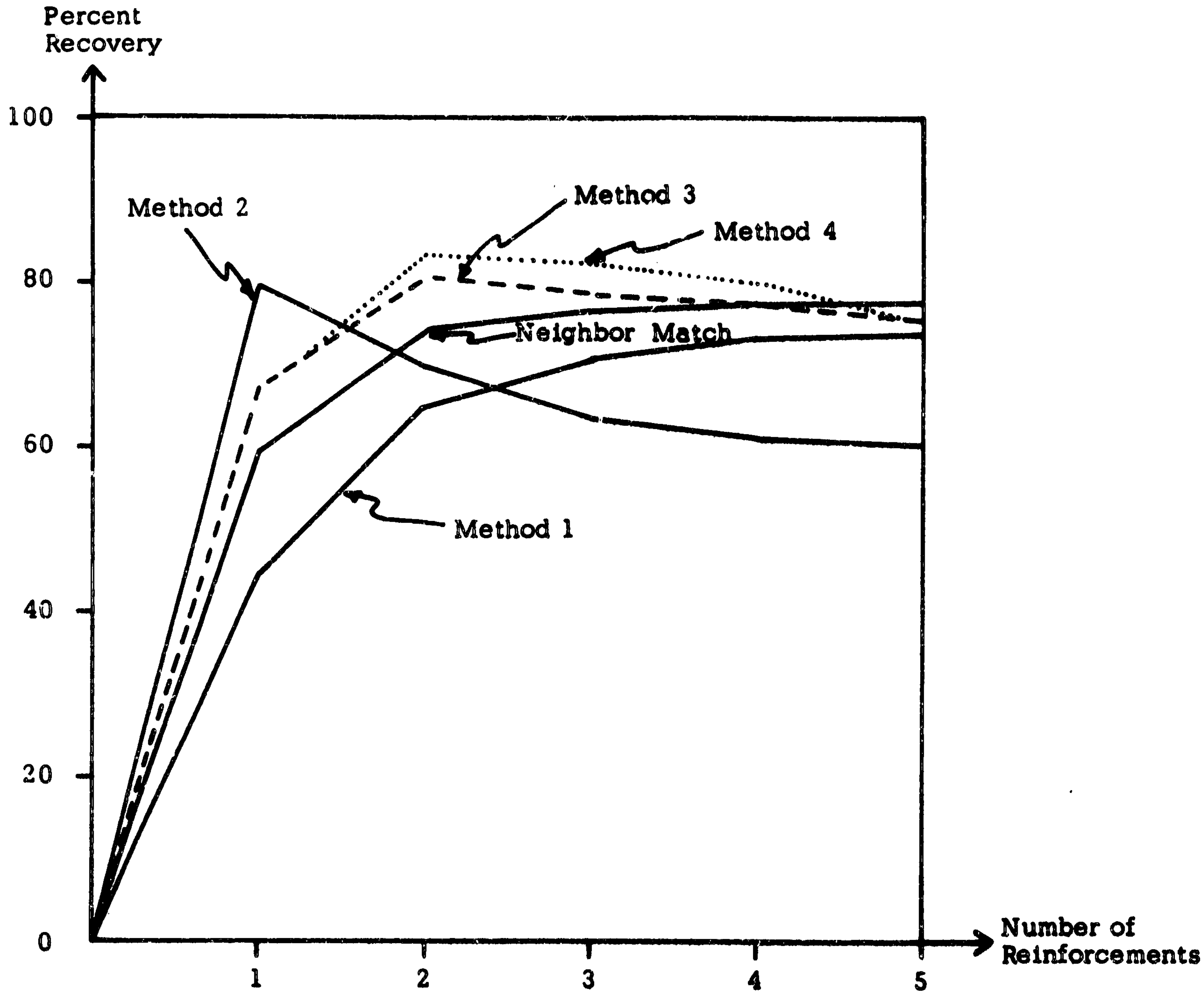


Figure 3.5. BAC-145 at 0°

IV. OPTIMAL CODING

4.1 Skeleton Coding

If a skeleton matrix is to be transmitted, it must first be coded in a suitable form. One way of coding the skeleton matrix is to form a code word for each non-zero point. With such a coding method, three pieces of information are required to describe each point: its vertical coordinate, its horizontal coordinate, and its resolution value. One of many methods may be used to describe these three characteristics, but the coded words must contain enough information for one to determine them accurately.

In transmitting an image, it may not be necessary to transmit each detail of the image. The receiver may desire an image of only a certain resolution, or, if a pattern recognition process is involved, the receiver may be able to recognize a pattern without information about each detail of the image. In either case, it would not be necessary to transmit every skeleton point.

Since it is not always required to transmit all skeleton points, it is desirable to arrange the skeleton point codes in an order such that the first points contain the largest possible area and that succeedingly less area is added with the addition of the area described by each succeeding skeleton point. One easy method of approximating such an ordering is to arrange the skeleton points in order of descending resolution. However, this method of ordering results in a poor approximation of the desired order.

A method of obtaining an order of code words such that each code word, as it is added, describes a maximum additional area is an optimum method of code word ordering. To implement such a method, a diamond described by one of the skeleton points with the largest resolution is placed on the matrix being formed, and its code word becomes the first code word to be transmitted. It is also placed on a matrix which will be called the trial matrix. Then, the diamond described by each remaining skeleton point is added, one by one, to the trial matrix. After each diamond is added, the total area of the region resulting from such an addition is calculated, and then the points on the trial matrix are reset to the values of the points on the matrix being formed. The code whose diamond addition yields the maximum area from the above trials becomes the second code word to be transmitted, and the process is repeated on the remaining code words to find the third code word, the fourth, and so on until all code words are arranged in the desired order. An example of such an ordering is shown in Figure 4.1 in which a rectangle similar to the one in Figure 1.4 is being formed. Figure 4.1 (a) shows the diamond resulting from the first code. The second diamond is added in Figure 4.1 (b), the third in Figure 4.1 (c), and the fourth in Figure 4.1 (d).

This method of ordering the skeleton code words provides the best obtainable ordering of code words for transmission. However, it is a time consuming method. A new matrix must be formed for each skeleton point tested before an optimum value can be found, and an optimum value must be found for each skeleton point to be transmitted. It is desirable to have a faster method in which the desired order of the code words is approximated.

4.2 Estimated Ordering

Each code word for a skeleton point has a weight associated with it. This weight gives information about the figure which the skeleton describes by telling how much area is occupied by the diamond which the code word describes. However, once a diamond has been placed on the matrix being formed from the transmitted code words, other diamonds near to this diamond may have areas which overlap with the area of the already placed diamond. In such a case, the full value of the weights

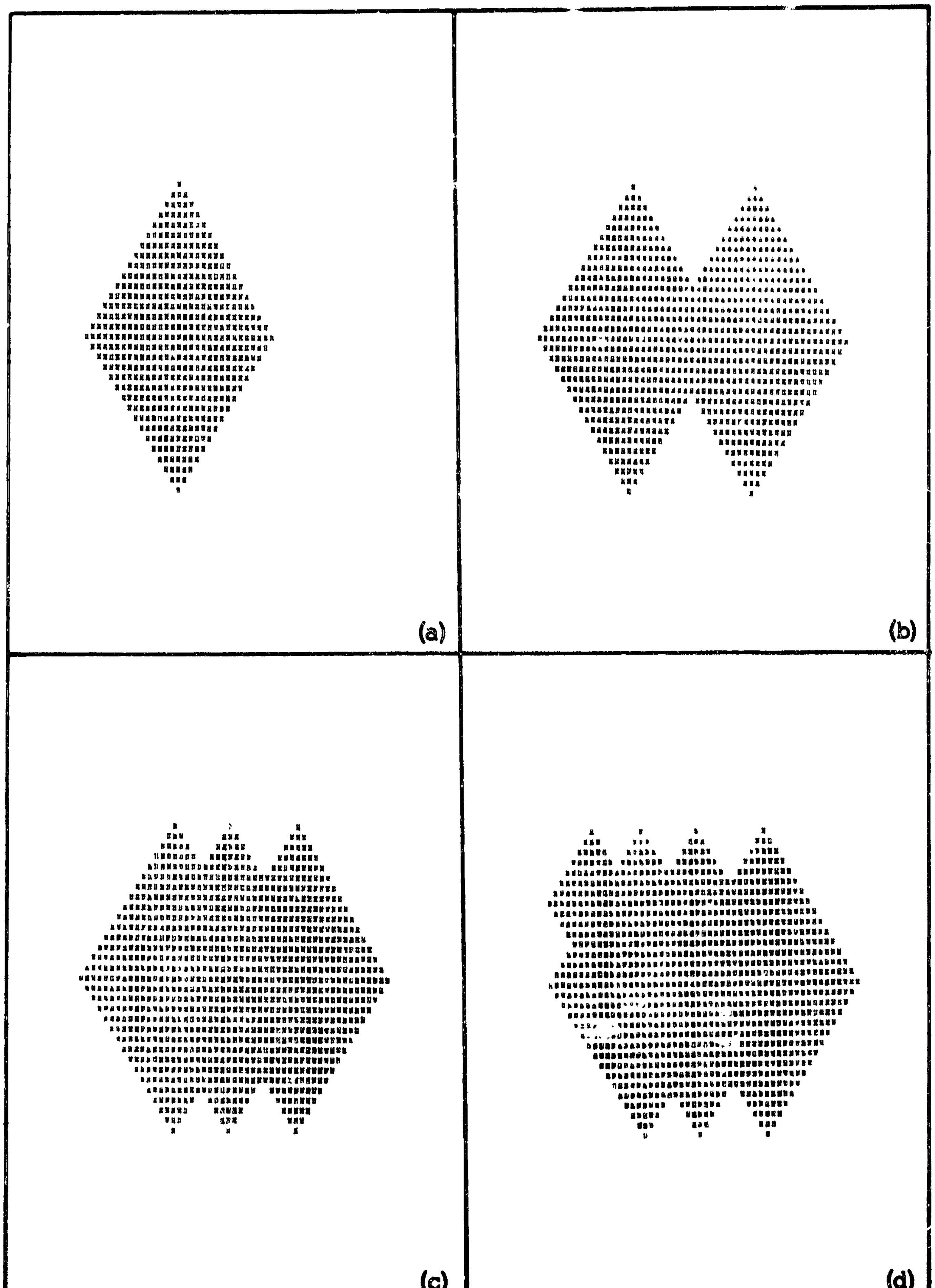


Figure 4.1

of the other diamonds is not gained as additional information about the figure being formed as these other diamonds are added, since some of the area described by the addition of these other diamonds is already described by the diamond which had been placed. Thus, the placement of any diamond on the matrix being formed will have an effect in determining how much information will be gained by the addition of other diamonds as they are added later.

In ordering code words for transmission, it is desirable to order the code words so that the greatest possible amount of information will be gained by the addition of the diamond described by each code word as it is added. The first step in such an ordering process is to determine the weight of each skeleton point and to select a point with the greatest weight as the first skeleton point to be transmitted. The diamond described by this skeleton point probably intersects some of the area described by some of the other skeleton points. In such cases, it is necessary to adjust the weights associated with the other skeleton points so that the weights will continue to provide an accurate measure of how much additional area would be described by the addition of the diamond associated with any skeleton point. After these adjustments are made, the code word describing the skeleton point with the largest weight is chosen as the second code word to be transmitted. The diamond described by this code word probably also intersects some of the diamonds described by other code words, so it is again necessary to make adjustments on the weights associated with the remaining code words before the next code word for transmission can be chosen. This process of choosing code words according to weights and making adjustments on the weights of the unchosen code words before the next code word is chosen is continued until all code words have been placed in the ordered sequence. Throughout the process, the weight associated with each unchosen skeleton point provides a measure of how much additional area would be described if the diamond associated with this skeleton point were added to the figure being formed.

The adjustment which should be made on each code word weight after a code word has been chosen for the transmission sequence is determined by the resolution of the chosen code word and the resolution of code word whose weight is being adjusted and by the distance between their skeleton points. The adjustment is made on the weight associated with each code word by approximating the percentage of area of the diamond described by this code word which intersects the diamond just chosen for the ordered sequence and then reducing the weight of the unchosen code word by this calculated percentage. The fraction of weight decrease, D, can be approximated as follows:

$$D = \begin{cases} (1/2)\ K\ (R_u - 1/2)/W_u & K > 0 \\ 0 & K \leq 0 \end{cases} \tag{4.1}$$

where W_u is the weight of the unchosen code word which is being adjusted, and K is the number of diagonal rows common to both diamonds and is given by

$$K = |R_u + R_c - 1| - |\Delta x| - |\Delta y| \tag{4.2}$$

where R_u is the resolution of the unchosen code word, R_c is the resolution of the code word just chosen, and Δx and Δy are the vertical and horizontal matrix distances between the skeleton points corresponding to the two code words being considered.

The number of rows that the two diamonds have in common, K, can be seen in equation 4.2 to increase as the resolutions of the diamonds increase, and to decrease as the distance between the diamonds increases. The number of rows, K, is multiplied by the average number of points in each row of the unchosen diamond, $R_u - 1/2$, to obtain an approximation of the total number of points in the rows involved in the diamond intersection. This result is multiplied by 1/2 to approximate the number of intersecting points since approximately one-half of the points in the common rows can be expected to be involved in the intersection of the diamonds. This result is divided by the total weight of the unchosen diamond, W_u, to determine D, an approximation of the fraction of the number of points in the unchosen diamonds which are involved in the intersection.

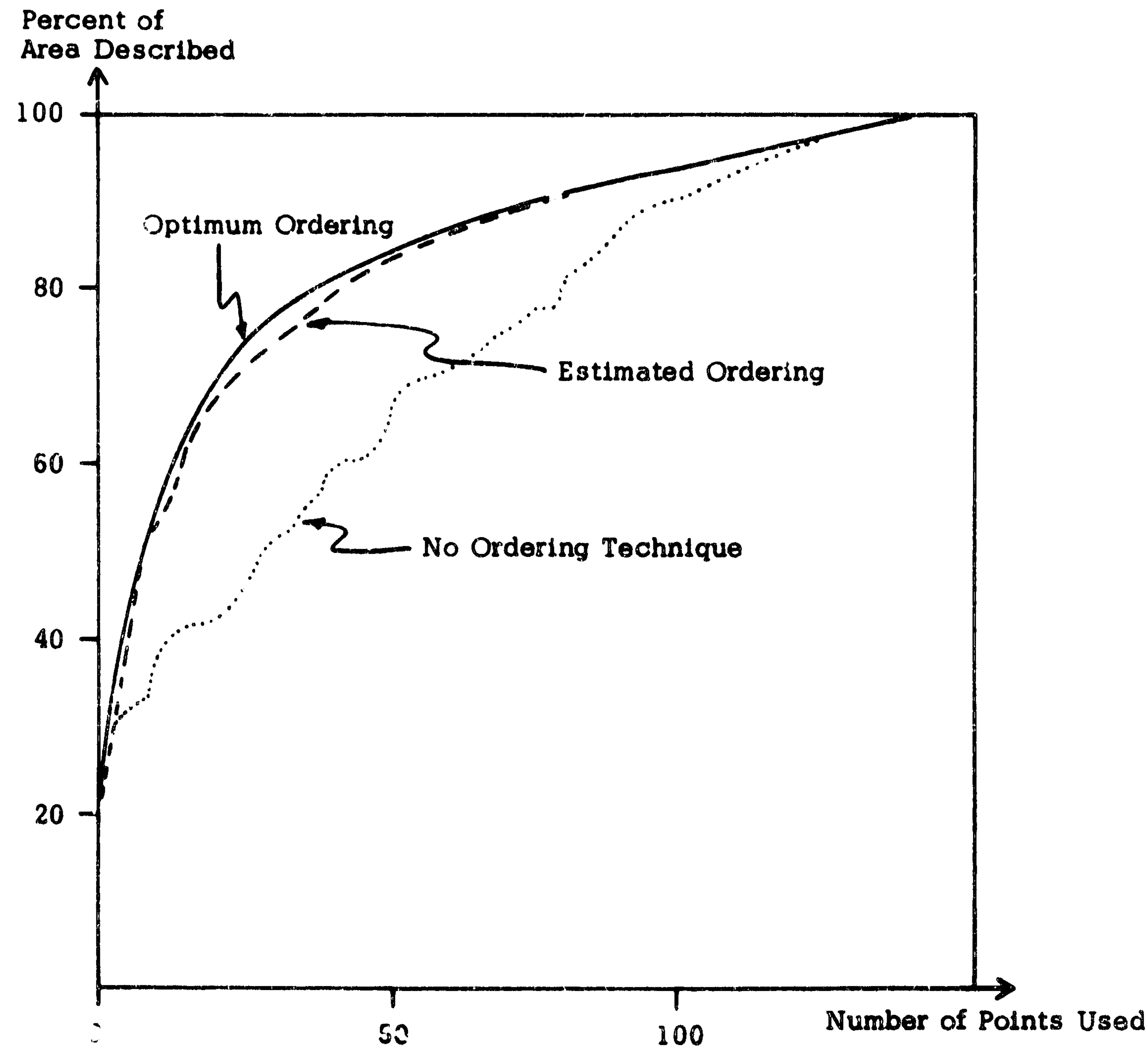
Using D as an intersection area estimator, the process of skeleton point code word ordering is a simple one. First, the weight of each skeleton point is calculated. The point with the largest weight becomes the first one chosen for transmission. Next, the weight of each remaining point is multiplied by $1 - D$. The skeleton point with the highest resulting weight is the next to be chosen for transmission. This process is repeated over and over until all points have been chosen.

The purpose of ordering points for transmission is to be able to transmit as much area with as few points as possible. A comparison of the estimated ordering method with the calculation of the optimum ordering method and no ordering technique at all is shown in Figure 4.2. The graph shows what percentage of the area is described by how many points where the points have been chosen in an order by the method designated in the graph. The figure used was a BAC-145 jet at 0°. The method of estimated ordering is shown to be nearly as effective in ordering codes for transmission as the "optimum" method. The estimated ordering method sequenced the codes twenty times faster.

The method of estimated ordering can also be applied to the square fitting transformation.

4.3 Boundary Description

The usefulness of the skeleton transform as a method of data reduction can be measured by comparing it with other methods of data reduction. A method of two-dimensional data reduction which is commonly used involves describing a figure by describing a path along its outside boundary. The path is described by a series of line segments, each one having a magnitude and a direction. For the case of the 60 x 60 matrix, a line segment can be as long as sixty units in any one of eight direc-



Percent of
Area Described
100
80
60
40
20
Optimum Ordering
Estimated Ordering
No Ordering Technique
0
50
100
Number of Points Used

Figure 4.2

tions. Thus, any line segment can be represented by nine bits of information. Each skeleton point can be represented by two coordinates and a magnitude of resolution. Such a point on a 60 x 60 matrix can be represented by seventeen bits.

It is difficult to compare the boundary description method and the skeleton method of data reduction since the effectiveness of both methods depend on the size and shape of the object being reduced and the skeleton method depends on orientation. Also, the skeleton method is intended for use when transmission of every detail of the object matrix is not required.

The skeleton method of data reduction is at least as effective as the boundary description method of data reduction. In some cases it is much better. For example, eighty percent of the area of the BAC-145 at 45° can be represented, using diamonds, by 238 bits with the skeleton method. 630 bits would be required for the same figure using the outlining method.

4.4 A Predictor Method

In transmission of video data, it is generally desirable to transmit only a portion of the data and to allow the receiver to predict the remainder. Such a method of data prediction can be applied to the skeleton method of data reduction using methods similar to the ones discussed in the previous chapter.

Use of the predictor method of data transmission requires that all points on the object matrix belong to a region. All examples which have been used so far involve two regions. One region includes the area on the object matrix in which the points have a value of one, and the other region has points of value zero. Skeleton points describing a certain percentage of each region are transmitted, and then the transmitted regions are allowed to propagate into the regions not described by the transmission in a manner similar to the propagation described for Method 2 of the noise reduction process.

An example of results of the prediction method of data transmission applied to a circle is shown in Figure 4.3. The object matrix is shown in Figure 4.3 (a). The skeleton matrix transformation of each region was taken using the square fitting method. Then, eighty percent of the area of each region was selected for transmission using the estimated ordering method. The selected regions are shown in Figure 4.3 (b). The zero region is shown by 0's and the blank area represents unselected points. The result obtained by the prediction method is shown in Figure 4.3 (c).

450 bits were required to transmit the circle using the predictor method, and a large number of errors were made. 612 bits were required to transmit the circle with no errors using the boundary description method. Since similar results were obtained with other figures, the predictor method of data reduction described apparently does not provide a satisfactory method of data reduction when only two regions are used.

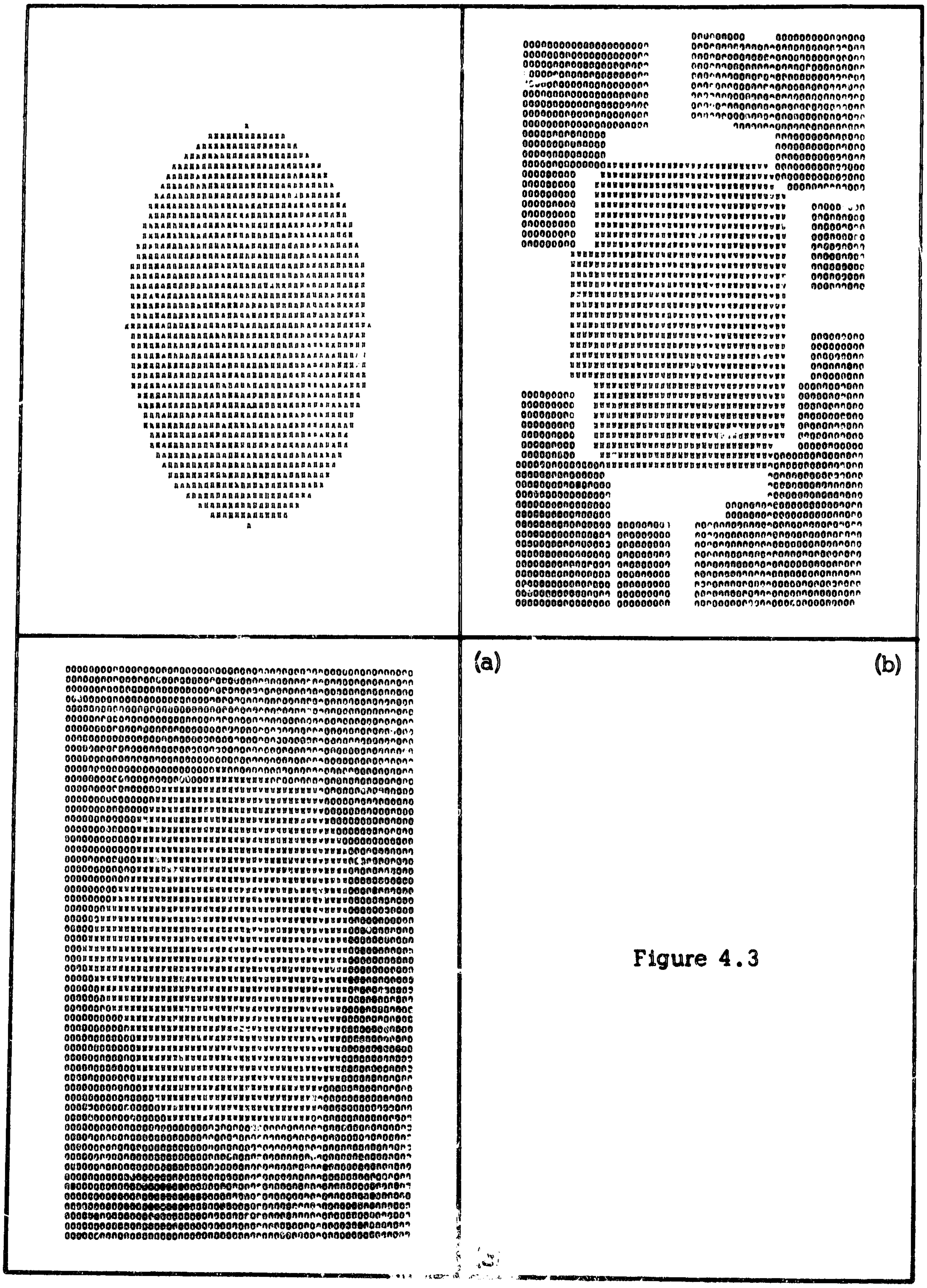


Figure 4.3

V. A PHOTOGRAPHIC EXAMPLE

5.1 Data

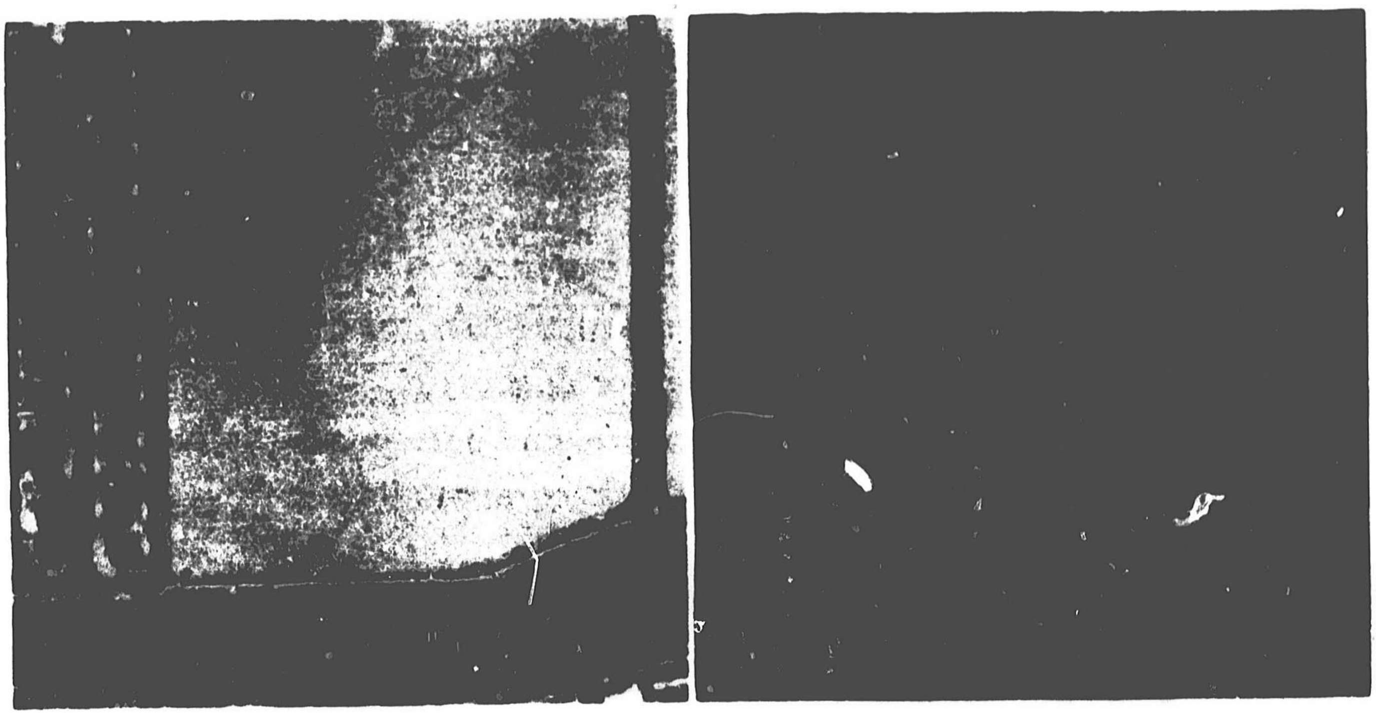
Most video data which will be enhanced and transmitted will be photographic imagery. Such data is generally remote sensory data obtained from reconnaissance planes or spacecraft. It is desirable, then, to test the methods of noise reduction and data compression discussed in previous chapters using actual photographic data.

Pre-processed photographic data was used as input data to test the skeleton transformation data enhancement and data reduction methods. Pre-processing of the data was done by Dalke[4] using Baye's statistical decision strategy to classify each point of an image into one of three categories. The data used as input to Dalke's processing technique consisted of three 70-millimeter transparencies selected from a set taken by an Itek nine-lens multi-spectral camera during a flight over Phoenix, Arizona, in July, 1965. The selected transparencies were sensitive to different visual regions.

The photographic density at 6400 corresponding points on each image was measured with a densitometer. These points were selected on an 80 x 80 grid with a .025-inch grid point spacing. The densitometer spot size was defocused so that the measured density was an average over a .025-inch diameter spot. The resulting measurements were entered into IBM cards in a format that designated their original location on the image.

The coded images were processed using an IBM 7040 computer to classify each point into one of three regions using Baye's decision theory. A 60 x 60 grid was chosen from the resulting image and coded onto IBM cards. This 60 x 60 grid was used as input data to examine the enhancement and data reduction methods discussed in this paper.

A positive print of the area included in the 60 x 60 grid of one of the photographic images is shown in Figure 5.1 (a). For reference in evaluating the results, the image in Figure 5.1 (a) was reproduced out



(a) (b)

Figure 5.1

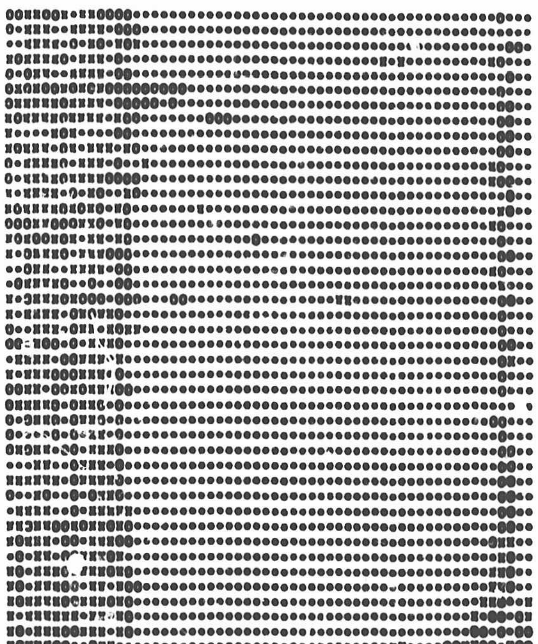


Figure 5.2

of focus. This de-focused image is shown in Figure 5.1 (b) and was obtained with sufficient care so that it represents the data as seen by the computer for Dalke's program. The computer output from Dalke's program which was used as computer input for the programs discussed here is shown in Figure 5.2.

5.2 Results

Noise reduction methods were applied to the object matrix shown in Figure 5.2. The methods used were the same as the ones described in Chapter 3 except that there were three regions instead of two. The results of Methods 1, 2, 3, and 4 with three reinforcements are shown in Figures 5.3 (a), (b), (c), and (d) respectively. It should be noted that 616 skeleton points were required to describe the result of Method 1 while the others required less than 300 points each.

The code words resulting from the noise reduction methods were ordered, region by region, using the estimated ordering method for square fitting discussed in the previous chapter. Use of the estimated ordering method permitted eighty percent of the area of the image to be described with less than sixty skeleton points when the image used was obtained from three reinforcement noise reductions using Methods 2, 3, or 4. This means that eighty percent of the image can be described by less than 1020 bits as compared to 7200 bits which would be required if the image grid was coded point by point.

The predictor method of transmitted image regeneration was applied to the noise reduced images using eighty percent of the image area for transmission. The transmitted areas for Methods 1, 2, 3, and 4 are shown in Figures 5.4 (a), (b), (c), and (d) respectively. The images reformed from the transmitted areas using the predictor method are shown in Figure 5.5.

The images in Figure 5.5 represent a reasonable approximation of the regions described by Figure 5.2. This would indicate that the pre-

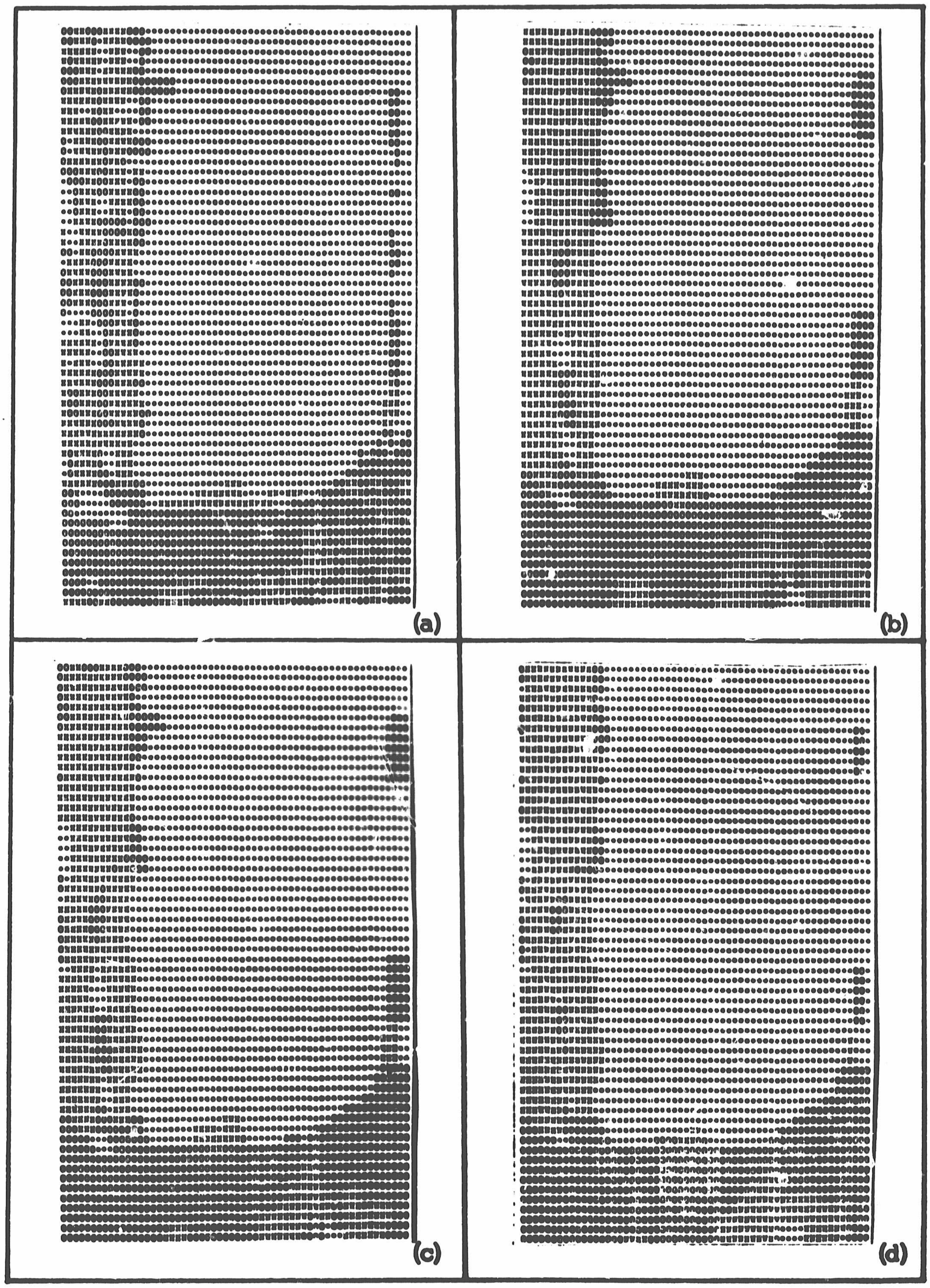


Figure 5.3

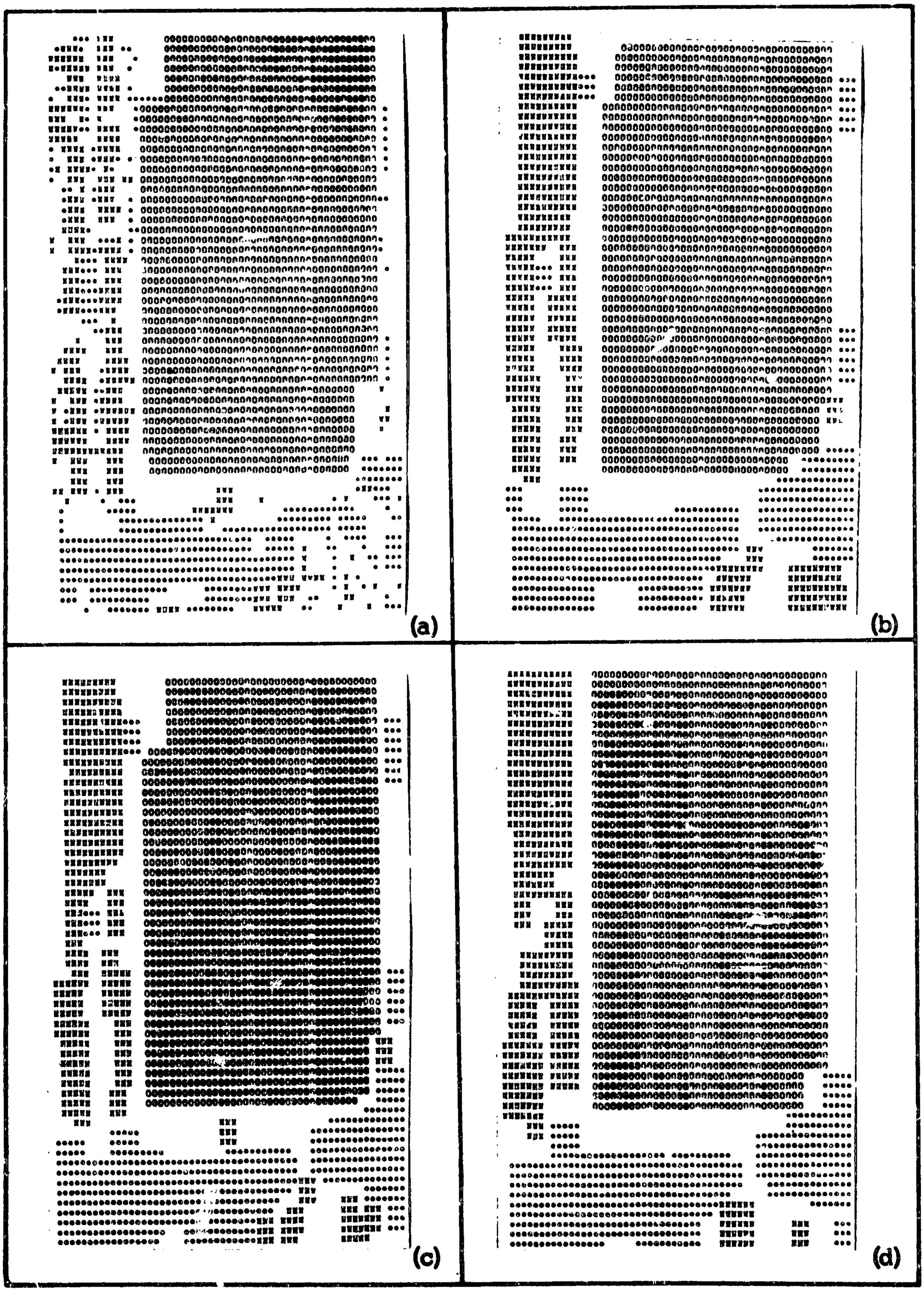


Figure 5.4

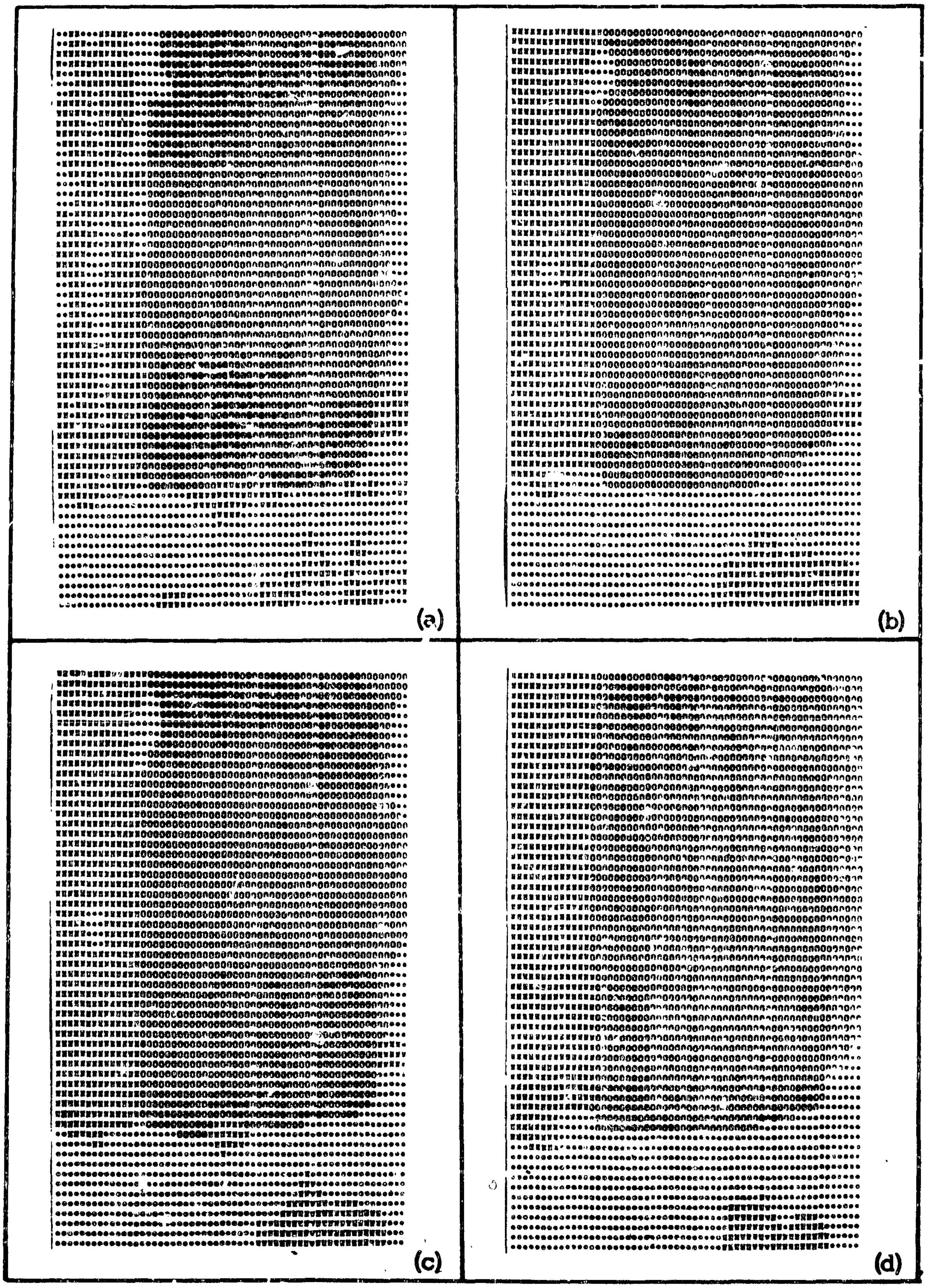


Figure 5.5

dictor method of transmitted image reformation described in Chapter 4 might be useful when more than two regions are used. In any case, the method appears to be useful in describing the imagery used here.

VI. CONCLUSIONS

6.1 Results

The two-dimensional medial point skeleton transformation described in the introduction provides a useful means of representing two-dimensional data such as radar imagery and video data. Regions of the image are described as the union of several diamonds, each of which is defined by a point on a skeleton matrix. The transformation is a useful one since the value of each skeleton point can easily be used to calculate the area of its corresponding diamond and therefore provides a direct indication of the importance of that point in the description of an image.

The use of the transformation for pattern recognition is restricted since introduction of a small amount of noise on an image being transformed greatly affects the resulting transformation. For this reason, the primary value of the skeleton transformation is its usefulness in data compression and noise reduction.

Since the transformation describes each image as the union of several straight-edged figures (diamonds or squares), the average size of the diamonds or squares described by the transformation is a function of the orientation of the image. This dependence is a strong one as shown with the example of the BAC-145 jet. The dependence can be used advantageously if it is suspected that the image has an orientation of its own by simply aligning the edges of the image to coincide with the edges of the diamonds or squares. The transformation can also be used to detect orientation of an image.

The skeleton transformation provides a useful method of noise reduction. Since noise generally occurs in video data as small isolated points or lines, it can be eliminated since the resolution associated with this noise is small and can be smoothed out. Noise was reduced quite effectively on the images used in Chapter 3, especially with Methods 2 and 4. These results suggest that a method of region growth where the regions are defined by points of high resolution obtained from the skeleton transformation provides a more effective method of noise reduction than does the more conventional neighbor search method.

The usefulness of the skeleton matrix transformation as a means of data reduction when only two regions are involved is questionable. It cannot be relied upon to provide a more effective means of data compression than the boundary description method. However, the skeleton matrix transformation provides an effective means of data reduction when more than two regions are involved. Its usefulness for this purpose is shown with the use of pre-processed photographic data. The data used contains only three regions. However, the methods used may be directly used to reduce data with as many regions as desired. It should be noted that the methods used treat each region separately. No adjustment is made for the relative similarity of the regions.

6.2 Recommendations for Further Investigation

A problem noted with the medial point skeleton transformation is that it does not provide a completely effective method of data compression. This shortcoming might be overcome by devising a method to describe the skeleton in a more general manner rather than point by point. Such a description might involve a two-dimensional least-squares curve fit of the more important skeleton points or a description of the skeleton shape using methods of syntax.[5] A study of a medial axis description of figures similar to a syntax description of the skeleton transformation has already been done by Philbrick.

The noise reduction methods described in Chapter 3 are only intended to suggest a few possible methods of noise reduction. Obviously many more ingenious methods can be contrived using the skeleton transformation. Best results would be expected using a region growth method.

The data reduction methods described involving more than two region make no adjustments to account for the relative similarity of the regions. However, for effective data reduction, it is desirable to classify similar regions together. In the example of the photographs taken of Phoenix, the merging of similar regions was accomplished by preprocessing the photographs with Dalke's Baye's Decision Theory program. However, it would definitely be advantageous to use a similarity of regions criteria to assign object matrix values during the noise reduction process. Such a process would be useful for direct processing of photographic imagery. Also, such a process might allow pattern recognition which uses both shape and similarity of regions as criteria.

APPENDIX A

Computer Subroutines

Most of the basic computer simulations done using the skeleton transformation were done in computer subroutines. Brief descriptions of some of these subroutines are given below to aid the reader in understanding the makeup of the computer programs used.

Subroutine SKEL finds the skeleton transformation of an object matrix using diamond fitting. Subroutine SSKEL is the same as SKEL except that it uses square fitting. Subroutine NSKEL uses SSKEL to find the skeleton matrix transformation of an object matrix with three regions.

Subroutine DIADD adds a diamond to a matrix given the matrix and a skeleton point code word. DIADD also finds the number of 1's in the matrix after the addition is made. Subroutine SDIADD is the same as DIADD except that squares are used instead of diamonds.

Subroutine PRNT prints a matrix printing blanks in place of zeros and X's in place of ones.

Subroutine OTLIN calculates the number of bits required to describe a figure using the boundary outline method.

Subroutine NCØDE finds the code words describing a skeleton matrix.

Subroutine DIAF orders code words using the "optimum" method.

Listings of these subroutines are shown on the following pages.

```
      SUBROU   E SMEL(IA,IS,NLARGE)
      DIMENS... IA(90,90),IB(90,90),IS(90,90)
      N=1
      DO 201 I=1,90
      DO 201 J=1,90
  201 IS(I,J)=0
      DO 202 I=1,90
      DO 202 J=1,90
  202 IB(I,J)=IA(I,J)
  203 DO 204 I=2,89
      DO 204 J=2,89
      IF(IA(I,J).EQ.1) GO TO 204
      IF(IA(I+1,J).EQ.1) IB(I+1,J)=2
      IF(IA(I-1,J).EQ.1) IB(I-1,J)=2
      IF(IA(I,J+1).EQ.1) IB(I,J+1)=2
      IF(IA(I,J-1).EQ.1) IB(I,J-1)=2
  204 CONTINUE
      DO 206 I=2,89
      DO 206 J=2,89
      IF(IB(I,J).NE.2) GO TO 206
      IF(IB(I+1,J).EQ.1) GO TO 205
      IF(IB(I-1,J).EQ.1) GO TO 205
      IF(IB(I,J+1).EQ.1) GO TO 205
      IF(IB(I,J-1).EQ.1) GO TO 205
      IS(I,J)=N
  205 IB(I,J)=0
  206 CONTINUE
      NLARGE=N
      N=N+1
      DO 207 I=1,90
      DO 207 J=1,90
  207 IA(I,J)=IB(I,J)
      DO 208 I=1,90
      DO 208 J=1,90
      IF(IA(I,J).EQ.1) GO TO 203
  208 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SSMEL(IA1,IA2,IB1,IB2,IS1,IS2,NLARGE)
      DIMENSION IA1(60,60),IA2(60,60),IB1(60,60),IB2(60,60),IS1(60,60),I
     1S2(60,60)
      N=1
      DO 251 I=1,60
      DO 251 J=1,60
      IS1(I,J)=0
  251 IS2(I,J)=0
      DO 252 I=1,60
      DO 252 J=1,60
C     DELETE THE FOLLOWING CARD IF BOTH IA1 AND IA2 ARE READ.
      IA2(I,J)=IA1(I,J)
      IB1(I,J)=IA1(I,J)
  252 IB2(I,J)=IA2(I,J)
  253 DO 255 I=1,59
      DO 255 J=1,59
      IF(IA2(I,J).EQ.1) GO TO 254
      IF(IA1(I,J).EQ.1) IB1(I,J)=2
      IF(IA1(I+1,J).EQ.1) IB1(I+1,J)=2
      IF(IA1(I,J+1).EQ.1) IB1(I,J+1)=2
      IF(IA1(I+1,J+1).EQ.1) IB1(I+1,J+1)=2
  254 IF(IA1(I+1,J+1).EQ.1) GO TO 255
      IF(IA2(I,J).EQ.1) IB2(I,J)=2
      IF(IA2(I+1,J).EQ.1) IB2(I+1,J)=2
      IF(IA2(I,J+1).EQ.1) IB2(I,J+1)=2
      IF(IA2(I+1,J+1).EQ.1) IB2(I+1,J+1)=2
  255 CONTINUE
      DO 259 I=1,59
      DO 259 J=1,59
      IF(IB2(I,J).NE.2) GO TO 257
      IF(IB1(I,J).EQ.1) GO TO 256
      IF(IB1(I+1,J).EQ.1) GO TO 256
      IF(IB1(I,J+1).EQ.1) GO TO 256
      IF(IB1(I+1,J+1).EQ.1) GO TO 256
      IS2(I,J)=N
  256 IB2(I,J)=0
  257 IF(IB1(I+1,J+1).NE.2) GO TO 259
      IF(IB2(I,J).EQ.1) GO TO 258
      IF(IB2(I+1,J).EQ.1) GO TO 258
      IF(IB2(I,J+1).EQ.1) GO TO 258
      IF(IB2(I+1,J+1).EQ.1) GO TO 258
      IS1(I+1,J+1)=N
  258 IB1(I+1,J+1)=0
  259 CONTINUE
      NLARGE=N
      N=N+1
      DO 260 I=1,60
      IB1(1,I)=0
      IB1(I,1)=0
      IB2(I,60)=0
  260 IB2(60,I)=0

      DO 261 I=1,60
      DO 261 J=1,60
      IA1(I,J)=IB1(I,J)
  261 IA2(I,J)=IB2(I,J)
      DO 262 I=1,59
      DO 262 J=1,59
      IF(IA1(I+1,J+1).EQ.1) GO TO 253
      IF(IA2(I,J).EQ.1) GO TO 253
  262 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NSMEL(IA1,IA2,IB1,IB2,IC1,IC2,IS1,IS2,NLARGE)
      DIMENSION IA1(60,60),IA2(60,60),IB1(60,60),IB2(60,60),IC1(60,60),I
     1C2(60,60),IS1(60,60),IS2(60,60)
      DO 152 I=1,60
      DO 152 J=1,60
      IC1(I,J)=0
      IC2(I,J)=0
      IF(IA1(I,J).EQ.1) GO TO 151
      IC1(I,J)=IA1(I,J)
      IA1(I,J)=0
  151 IF(IA2(I,J).EQ.1) GO TO 152
      IC2(I,J)=IA2(I,J)
      IA2(I,J)=0
  152 CONTINUE
      CALL SSMEL(IA1,IA2,IB1,IB2,IS1,IS2,NLARGE)
      DO 153 I=1,60
      DO 153 J=1,60
      IF(IC1(I,J).EQ.0) IC1(I,J)=IS1(I,J)+100
      IF(IC2(I,J).EQ.0) IC2(I,J)=IS2(I,J)+100
      IA1(I,J)=0
      IA2(I,J)=0
      IF(IC1(I,J).EQ.2) IA1(I,J)=1
  153 IF(IC2(I,J).EQ.2) IA2(I,J)=1
      CALL SSMEL(IA1,IA2,IB1,IB2,IS1,IS2,NN)
      IF(NN.GT.NLARGE)NLARGE=NN
      DO 154 I=1,60
      DO 154 J=1,60
      IF(IC1(I,J).EQ.2) IC1(I,J)=IS1(I,J)+200
      IF(IC2(I,J).EQ.2) IC2(I,J)=IS2(I,J)+200
      IA1(I,J)=0
      IA2(I,J)=0
      IF(IC1(I,J).EQ.4) IA1(I,J)=1
  154 IF(IC2(I,J).EQ.4) IA2(I,J)=1
      CALL SSMEL(IA1,IA2,IB1,IB2,IS1,IS2,NN)
      IF(NN.GT.NLARGE) NLARGE=NN
      DO 155 I=1,60
      DO 155 J=1,60
      IF(IC1(I,J).EQ.4) IC1(I,J)=IS1(I,J)+400
  155 IF(IC2(I,J).EQ.4) IC1(I,J)=IS1(I,J)+400
      DO 156 I=1,60
      IC1(1,I)=0
      IC1(I,1)=0
      IC2(I,60)=0
  156 IC2(60,I)=0
      RETURN
      END
```

```
      SUBROUTINE DIADD(IA,KOD,KVAL)
      DIMENSION IA(56,56)
      IW=KOD/10000
      II=(KOD-IW*10000)/100
      IJ=KOD-10000*IW-100*II
      IA(II,IJ)=1
      IF(IW.EQ.1) GO TO 302
      DO 301 K=2,IW
      KM1=K-1
      I2=II-KM1
      I3=II+KM1
      IA(I2,IJ)=1
      IA(I3,IJ)=1
      DO 301 L=1,KM1
      I4=I2+L
      I5=I3-L
      I6=IJ-L
      I7=IJ+L
      IA(I4,I6)=1
      IA(I4,I7)=1
      IA(I5,I6)=1
  301 IA(I5,I7)=1
  302 KVAL=0
      DO 303 I=2,55
      DO 303 J=2,55
  303 IF(IA(I,J).EQ.1) KVAL=KVAL+1
      RETURN
      END
```

```
      SUBROUTINE SDIADD(IA1,IA2,IC1,IC2,KOD,KVAL,NSUR)
      DIMENSION IA1(60,60),IA2(60,60),IC1(60,60),IC2(60,60)
      DO 301 I=1,60
      DO 301 J=1,60
  301 IC1(I,J)=0
      IW=KOD/100000
      II=(KOD-IW*100000)/1000
      IJ=(KOD-100000*IW-1000*II)/10
      K=2
      IF((KOD/2)*2.EQ.KOD) GO TO 394
      IC1(II,IJ)=1
      IF(IW.EQ.1) GO TO 398
  391 I1=II-(K/2)
      I2=II+((K+1)/2)-1
      I3=IJ-(K/2)
      I4=IJ+((K+1)/2)-1
      DO 392 N=I1,I2
      IA2(N,I3)=1
  392 IA2(N,I4)=1
      DO 393 N=I3,I4
      IA2(I1,N)=1
  393 IA2(I2,N)=1
      IF(IW.EQ.K) GO TO 398
      K=K+1
      GO TO 399
  394 IA2(II,IJ)=1
      IF(IW.EQ.1) GO TO 398
  399 I1=II-((K+1)/2)+1
      I2=II+K/2
      I3=IJ-((K+1)/2)+1
      I4=IJ+K/2
      DO 396 N=I1,I2
      IC1(N,I3)=1
  396 IC1(N,I4)=1
      DO 397 N=I3,I4
      IC1(I1,N)=1
  397 IC1(I2,N)=1
      IF(IW.EQ.K) GO TO 398
      K=K+1
      GO TO 391
  398 IF((KOD/2)*2.EQ.KOD) GO TO 359
      DO 302 I=1,60
      DO 302 J=1,60
      IF(IC1(I,J)+IC2(I,J).EQ.2) NSUR=NSUR+1
  302 IF(IC1.EQ.1) IA1(I,J)=1
  359 KVAL=0
      DO 300 I=1,60
      DO 300 J=1,60
      IF(IA1(I,J).EQ.1) KVAL=KVAL+1
  300 IF(IA2(I,J).EQ.1) KVAL=KVAL+1
      RETURN
```

```
      SUBROUTINE DIAF(IA,IB,KODE,KODO,NCO,KVAL)
      DIMENSION IA(60,60),IB(60,60),KODE(500)
      ILL=0
      DO 402 N=1,NCO
      IF((((KODE(N)/10000)**2)+((KODE(N)/10000-1)**2)).LE.ILL) GO TO 402
      DO 401 I=2,59
      DO 401 J=2,59
  401 IB(I,J)=IA(I,J)
      KOD=KODE(N)
      CALL DIADD(IB,KOD,KVALN)
      IF(KVALN-KVAL.LE.ILL) GO TO 402
      ILL=KVALN-KVAL
      KODO=N
  402 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRNT(IA,IC)
      DIMENSION IA(56,56),IC(56,56)
      DATA IBLANK/1H /,IX/1HX/
      DO 501 I=1,56
      DO 501 J=1,56
      IF(IA(I,J).EQ.0) IC(I,J)=IBLANK
      IF(IA(I,J).EQ.1) IC(I,J)=IX
  501 CONTINUE
      WRITE(6,502)((IC(I,J),J=1,56),I=1,56)
  502 FORMAT(12X,56A1)
      RETURN
      END
```

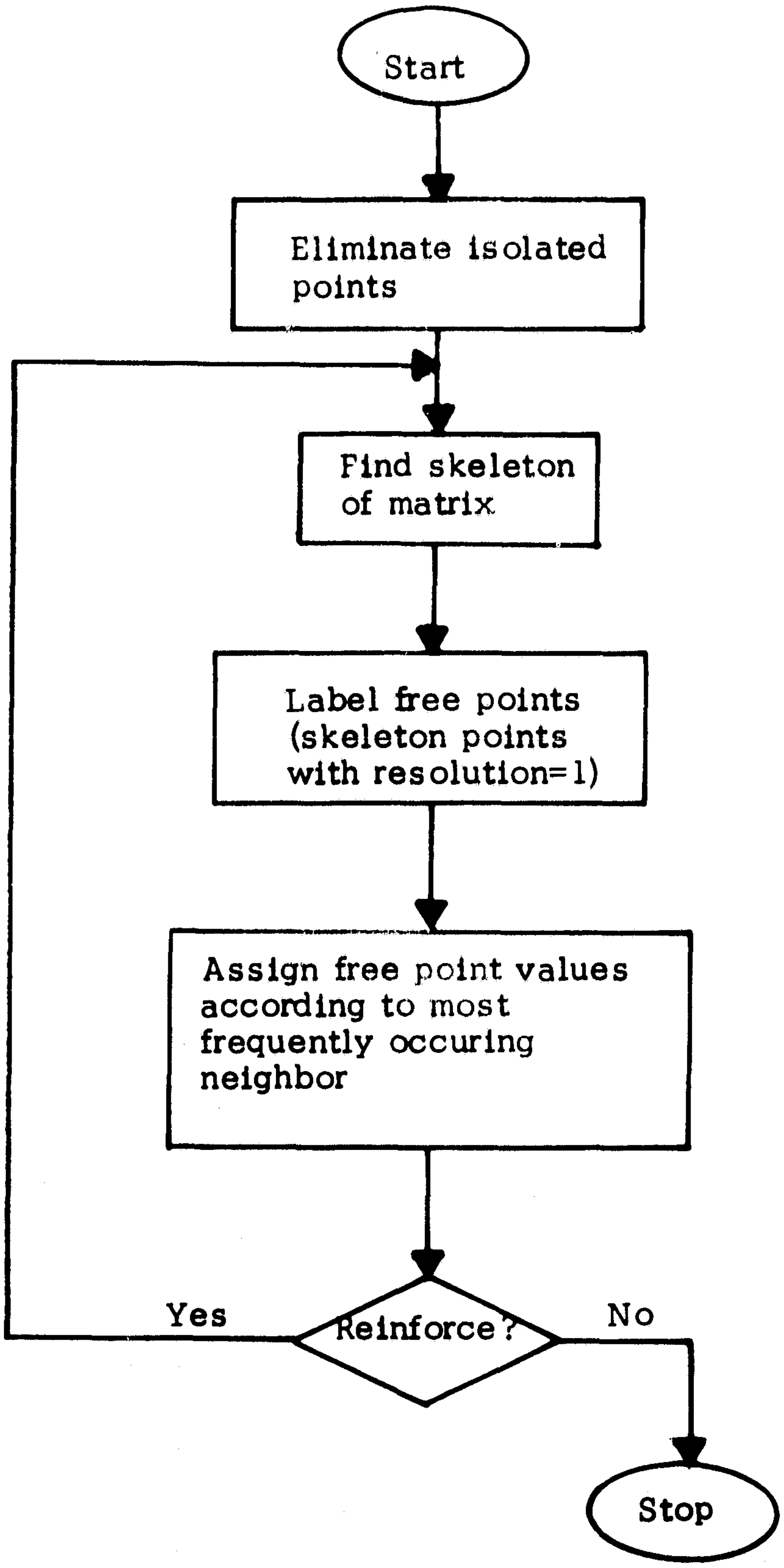
```
      SUBROUTINE OTLIN(IA,IB,KODE,JE)
C     MUST APPEAR BEFORE SMEL IF USED
      DIMENSION IA(96,96),IB(96,96),KODE(200),JE(8)
      DO 601 I=1,96
      DO 601 J=1,96
  601 IB(I,J)=0
      DO 602 I=1,96
      DO 602 J=1,96
      IF(IA(I,J).EQ.1) GO TO 603
  602 CONTINUE
  603 IB(I,J)=3
      N=0
  604 N=N+1
      IF(IA(I,J+1)*10+IA(I+1,J+1).NE.1) GO TO 605
      KODE(N)=1
      I=I+1
      J=J+1
      GO TO 612
  605 IF(IA(I+1,J+1)*10+IA(I+1,J).NE.1) GO TO 606
      KODE(N)=2
      I=I+1
      GO TO 612
  606 IF(IA(I+1,J)*10+IA(I+1,J-1).NE.1) GO TO 607
      KODE(N)=3
      I=I+1
      J=J-1
      GO TO 612
  607 IF(IA(I+1,J-1)*10+IA(I,J-1).NE.1) GO TO 608
      KODE(N)=4
      J=J-1
      GO TO 612
  608 IF(IA(I,J-1)*10+IA(I-1,J-1).NE.1) GO TO 609
      KODE(N)=5
      I=I-1
      J=J-1
      GO TO 612
  609 IF(IA(I-1,J-1)*10+IA(I-1,J).NE.1) GO TO 610
      KODE(N)=6
      I=I-1
      GO TO 612
  610 IF(IA(I-1,J)*10+IA(I-1,J+1).NE.1) GO TO 611
      KODE(N)=7
      I=I-1
      J=J+1
      GO TO 612
  611 KODE(N)=8
      J=J+1
  612 IF(IB(I,J).NE.3) GO TO 604
      WRITE(6,613)
  613 FORMAT(14H1OUTLINE CODES)
      KODE(N+1)=KODE(1)


      NBITS=0
      L=1
      DO 616 K=1,N
      IF(KODE(K).NE.KODE(K+1)) GO TO 614
      L=L+1
      GO TO 616
  614 KU=KODE(K)
      JD=JE(KU)
      WRITE(6,615) L,JD
  615 FORMAT(3X,I2,A2)
      L=1
      NBITS=NBITS+9
  616 CONTINUE
      WRITE(6,617) NBITS
  617 FORMAT(10H NO. BITS=I4)
      RETURN
      END
```

```
      SUBROUTINE NCODE(IB,N,KODE)
      DIMENSION IB(90,90),KODE([illegible])
      N=0
      DO 101 I=2,95
      DO 101 J=2,95
      IF(IB(I,J).EQ.0) GO TO 101
      N=N+1
      KODE(N)=IB(I,J)*10000+I*100+J
  101 CONTINUE
      RETURN
      END
```

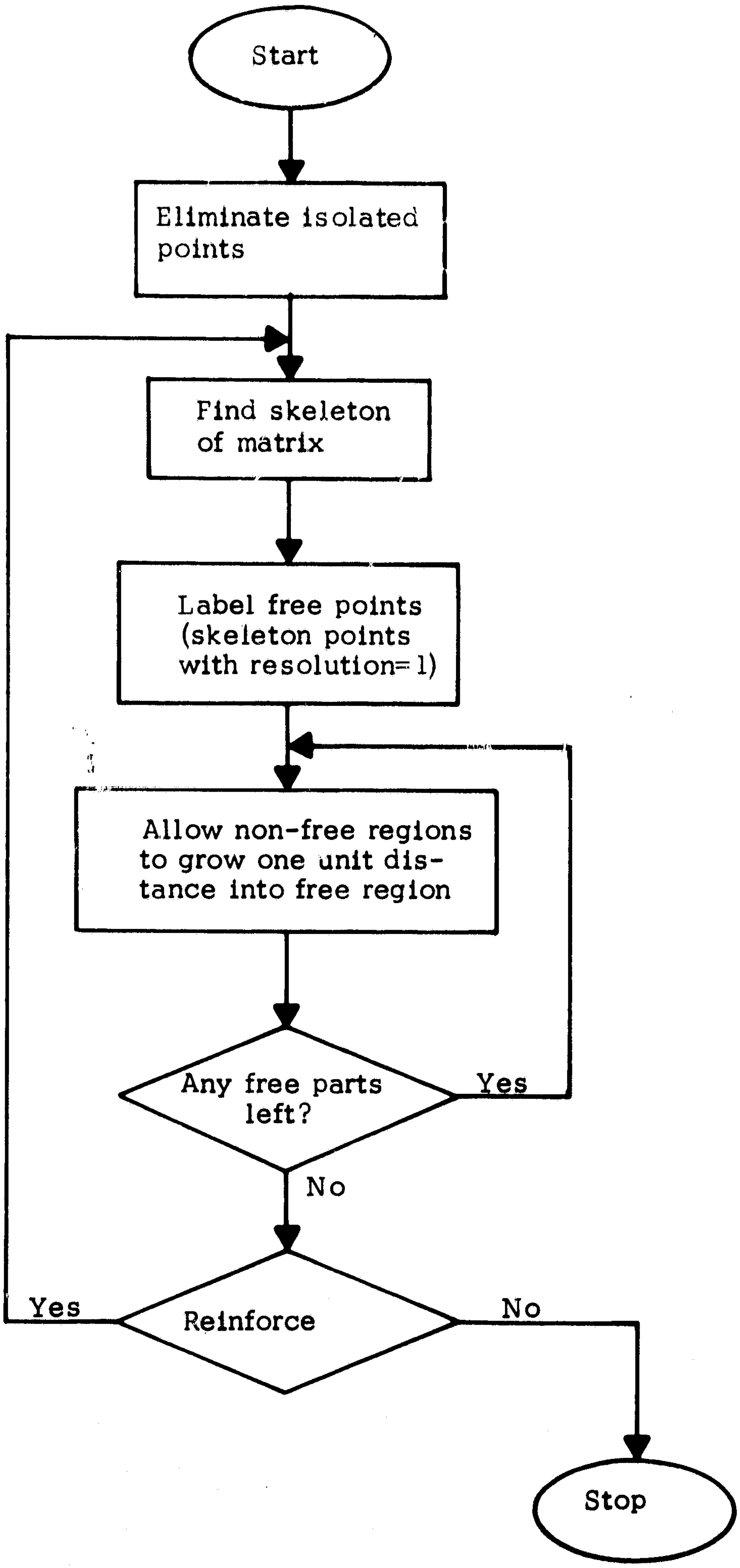
APPENDIX B

Flow Charts for Noise Reduction Methods

Method 1

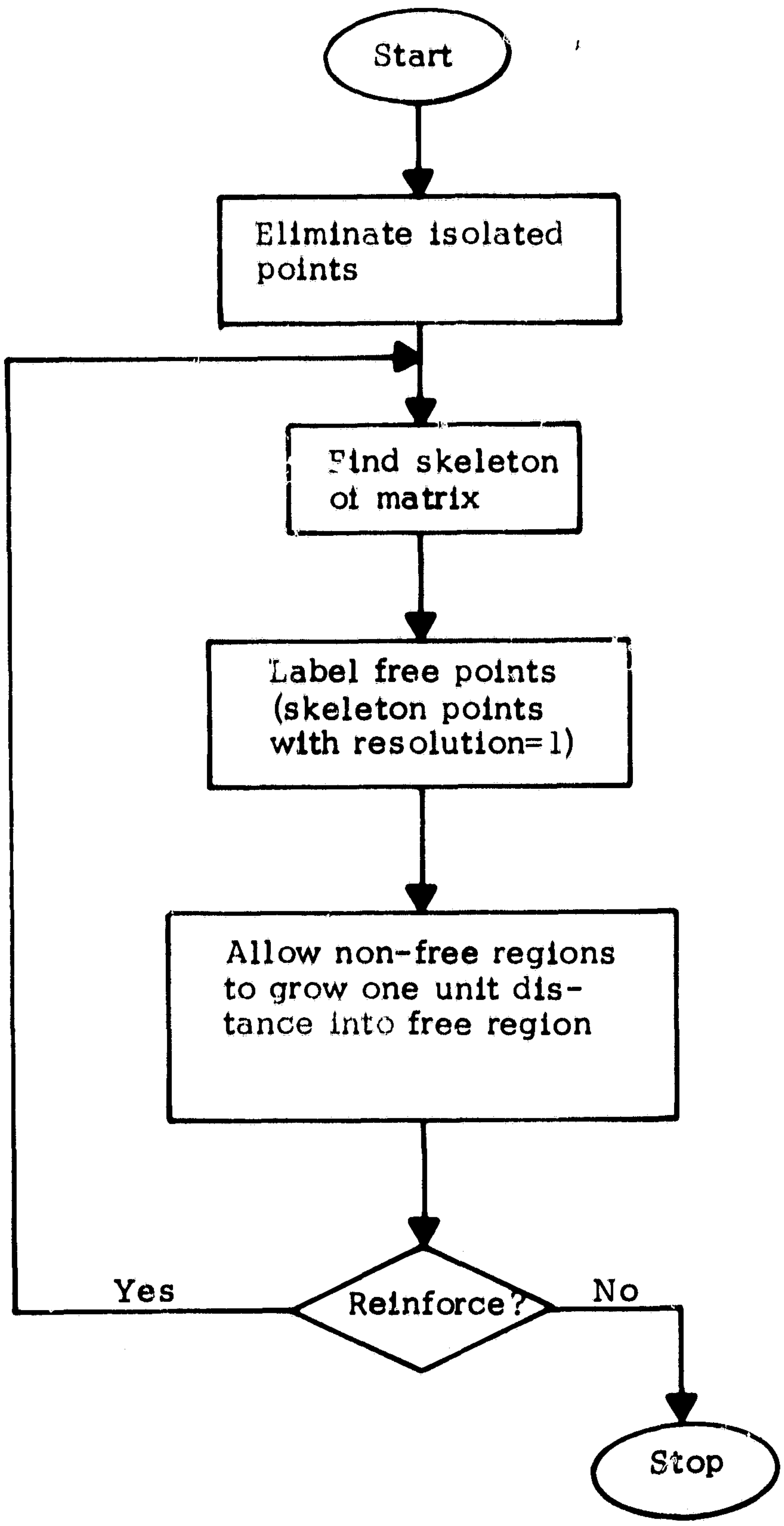


Method 2

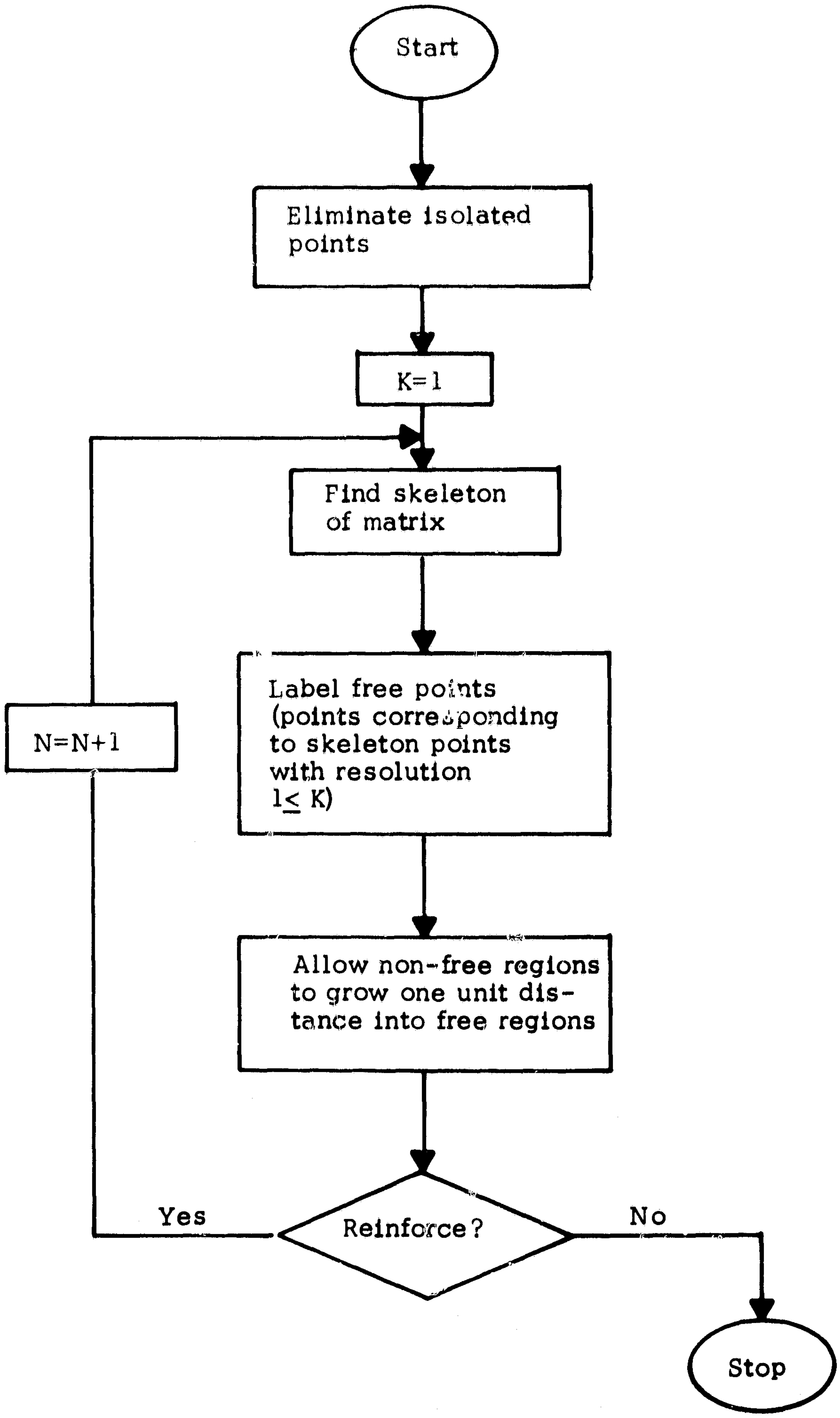


Start
Eliminate isolated points
Find skeleton of matrix
Label free points (skeleton points with resolution= 1)
Allow non-free regions to grow one unit distance into free region
Any free parts left?
Yes
No
Reinforce
Yes
No
Stop

Method 3


Start
Eliminate isolated points
Find skeleton of matrix
Label free points (skeleton points with resolution=1)
Allow non-free regions to grow one unit distance into free region
Reinforce?
Yes
No
Stop

Method 4


Start
Eliminate isolated points
K=1
Find skeleton of matrix
Label free points (points corresponding to skeleton points with resolution 1≤ K)
Allow non-free regions to grow one unit distance into free regions
Reinforce?
Yes
No
N=N+1
Stop

REFERENCES

1. Blum, Harry. "On Associated Machines for Dealing with the Visual Field and Some of its Biological Implications"; Air Force Cambridge Research Laboratory, Bedford, Massachusetts, February 1962.

2. Kotelly, John Christopher. "A Mathematical Model of Blum's Theory of Pattern Recognition"; Air Force Cambridge Research Laboratories, L. G. Hanscom Field, Massachusetts, April 1963.

3. Philbrick, Otis. "A Study of Shape Recognition Using the Medial Axis Transformation"; Northeastern University, May 1966.

4. Dalke, George W. "Automatic Processing of Multi-Spectral Images"; Center for Research, Inc., University of Kansas, Lawrence, Kansas, 1966.

5. Narasimham, R. "A Linguistic Approach to Pattern Recognition"; Digital Computer Laboratory, University of Illinois, Urbana, Illinois, July 1962.